ICES REPORT   11-32

October 2011

Adaptive Hessian-based Non-stationary Gaussian Process Response Surface Method for Probability Density Approximation with Application to Bayesian Solution of Large-scale Inverse Problems

by

Tan Bui-Thanh and Omar Ghattas and David Higdon

Report Documentation Page

1. REPORT DATE **OCT 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Adaptive Hessian-based Non-stationary Gaussian Process Response Surface Method for Probability Density Approximation with Application to Bayesian Solution of Large-scale Inverse Problems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Texas at Austin,Institute for Computational Engineering and Sciences,Austin,TX,78712**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We develop an adaptive Hessian-based non-stationary Gaussian process response surface method to approximate a probability density function (pdf) that exploits its structure, in particular the Hessian of its negative logarithm. Of particular interest to us are pdfs that arise from the Bayesian solution of large-scale inverse problems, which imply very expensive-to-evaluate pdfs. The method can be considered as a piecewise adaptive Gaussian approximation in which a Gaussian tailored to the local Hessian of the negative log probability density is constructed for each sub- region in high dimensional parameter space. The task of e ciently partitioning the parameter space into sub-regions is done implicitly through Hessian-informed membership probability functions. The Gaussian process machinery is then employed to glue all local Gaussian approximations into a global analytical response surface that is far cheaper to evaluate than the original expensive probability density. The resulting response surface is also equipped with an analytical variance estimate that can be used to assess the uncertainty of the approximation. One of the key components of our proposed approach is an adaptive sampling strategy for exploring the parameter space e ciently during the computer experimental design step, which aims to nd training points with high probability density. The detailed construction and an analysis of the method are presented. We then demonstrate the accuracy and e ciency of the proposed method on several example problems, including inverse shape electromagnetic scattering in 24-dimensional parameter space.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**36**	

ADAPTIVE HESSIAN-BASED NON-STATIONARY GAUSSIAN PROCESS RESPONSE SURFACE METHOD FOR PROBABILITY DENSITY APPROXIMATION WITH APPLICATION TO BAYESIAN SOLUTION OF LARGE-SCALE INVERSE PROBLEMS

TAN BUI-THANH [†], OMAR GHATTAS [†‡§], AND DAVID HIGDON [¶]

Abstract. We develop an adaptive Hessian-based non-stationary Gaussian process response surface method to approximate a probability density function (pdf) that exploits its structure, in particular the Hessian of its negative logarithm. Of particular interest to us are pdfs that arise from the Bayesian solution of large-scale inverse problems, which imply very expensive-to-evaluate pdfs. The method can be considered as a piecewise adaptive Gaussian approximation in which a Gaussian tailored to the local Hessian of the negative log probability density is constructed for each sub-region in high dimensional parameter space. The task of efficiently partitioning the parameter space into sub-regions is done implicitly through Hessian-informed membership probability functions. The Gaussian process machinery is then employed to glue all local Gaussian approximations into a global analytical response surface that is far cheaper to evaluate than the original expensive probability density. The resulting response surface is also equipped with an analytical variance estimate that can be used to assess the uncertainty of the approximation. One of the key components of our proposed approach is an adaptive sampling strategy for exploring the parameter space efficiently during the computer experimental design step, which aims to find training points with high probability density. The detailed construction and an analysis of the method are presented. We then demonstrate the accuracy and efficiency of the proposed method on several example problems, including inverse shape electromagnetic scattering in 24-dimensional parameter space.

Key words. probability density approximation; Gaussian process; response surface; adaptive sampling; computer experimental design; non-stationary; curse of dimensionality; Bayesian inversion; covariance function; membership probability; adjoint; Hessian.

AMS subject classifications. 62G07, 62G08, 62K20

1. Introduction. Solving large-scale ill-posed inverse problems that are governed by partial differential equations (PDEs) is both of great practical importance in science and industry as well as tremendously challenging. Classical deterministic inverse methodologies, which provide point estimates of the solution, are not capable of rigorously accounting for the uncertainty in the inverse solution. The Bayesian formulation provides a systematic quantification of uncertainty by posing the inverse problem as one of statistical inference. The Bayesian framework for inverse problems proceeds as follows: given observational data and their uncertainty, the governing forward problem and its uncertainty, and a prior probability density function (pdf) describing uncertainty in the parameters $\mathbf{m} \in \mathbb{R}^N$, the solution of the inverse problems is the posterior probability distribution $\pi_{\text{post}}(\mathbf{m})$ over the parameters. Bayes' Theorem explicitly gives the posterior pdf as

$$\pi_{\text{post}}(\mathbf{m}|\mathbf{y}_{\text{obs}}) \propto \pi_{\text{prior}}(\mathbf{m})\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m}),$$

which combines the prior pdf $\pi_{\text{prior}}(\mathbf{m})$ and the likelihood $\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m})$. The prior encodes any knowledge or assumptions about the parameter space that we may wish

---

[†]Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712, USA.

[‡]Jackson School of Geosciences, The University of Texas at Austin, Austin, TX 78712, USA.

[§]Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, USA.

[¶]Statistical Sciences, CCS-6, Los Alamos National Laboratory, Los Alamos, NM, 87545 USA

to impose before any data are considered, while the likelihood $\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m})$ explicitly represents the probability that a given set of parameters $\mathbf{m}$ might give rise to the observed data $\mathbf{y}_{\text{obs}} \in \mathbb{R}^p$. For simplicity of exposition, we assume that the prior is Gaussian and that the measurement and PDE model errors are combined into a noise term $\mathbf{e} = \mathbf{y}_{\text{obs}} - \mathbf{f}(\mathbf{m})$, which is additive and i.i.d. Gaussian. Then the pdf's for the prior and likelihood can be written in the form

$$\pi_{\text{prior}}(\mathbf{m}) \propto \exp\left(-\frac{1}{2}(\mathbf{m} - \bar{\mathbf{m}}_{\text{prior}})^T \mathbf{\Gamma}_{\text{prior}}^{-1}(\mathbf{m} - \bar{\mathbf{m}}_{\text{prior}})\right),$$

$$\pi_{\text{like}}(\mathbf{e}) \propto \exp\left(-\frac{1}{2}(\mathbf{e} - \bar{\mathbf{e}})^T \mathbf{\Gamma}_{\text{noise}}^{-1}(\mathbf{e} - \bar{\mathbf{e}})\right),$$

respectively, where $\bar{\mathbf{m}}_{\text{prior}}$ is the mean of the prior distribution, $\bar{\mathbf{e}}$ the mean of the Gaussian noise, $\mathbf{\Gamma}_{\text{prior}} \in \mathbb{R}^{N \times N}$ the covariance matrix for the prior, and $\mathbf{\Gamma}_{\text{noise}} \in \mathbb{R}^{p \times p}$ the covariance matrix of the noise. Restating Bayes' theorem with these Gaussian pdf's, we find that

$$\pi_{\text{post}}(\mathbf{m}) \propto \exp\left(-\frac{1}{2}\|\mathbf{m} - \bar{\mathbf{m}}_{\text{prior}}\|^2_{\mathbf{\Gamma}_{\text{prior}}^{-1}} - \frac{1}{2}\|\mathbf{y}_{\text{obs}} - \mathbf{f}(\mathbf{m}) - \bar{\mathbf{e}}\|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}}\right), \qquad (1.1)$$

where $\mathbf{f}(\mathbf{m})$ is the (nonlinear) operator mapping parameters to observations. Note that the seemingly simple expression $\mathbf{f}(\mathbf{m})$ belies the complexity of the underlying computations, which involves: (i) creation of the PDE model for given parameters $\mathbf{m}$; (ii) solution of the governing PDEs to yield the output state variables; and (iii) extraction of the observables (i.e., the values of the states at the observation locations in space and time). In general, $\mathbf{f}(\mathbf{m})$ is nonlinear, even when the forward PDEs are linear in the state variables, since the parameters couple with the states nonlinearly in the forward PDEs.

As is clear from the expression (1.1), despite the choice of prior and noise probability distributions as Gaussian, the posterior probability distribution need not be Gaussian, due to the nonlinearity of $\mathbf{f}(\mathbf{m})$. The non-Gaussianity of the posterior poses challenges for computing statistics for typical large-scale inverse problems since π_{post} is often a surface in high (thousands or millions) dimensions, and evaluating each point on this surface requires a solution of the forward PDEs. Numerical quadrature to compute the mean and covariance matrix, for example, is out of the question. Usually, the method of choice for computing statistics is Markov chain Monte Carlo (MCMC) [28], which judiciously samples the posterior distribution, so that sample statistics can be used to approximate the exact ones. But the use of MCMC for large-scale inverse problems is still prohibitive for expensive forward problems and high dimensional parameter spaces, since even for modest numbers of parameters, the number of samples required can be in the thousands or millions. Nevertheless, MCMC can be more efficient by exploiting higher order information such as the Hessian [44].

Since solving the forward PDEs is the most expensive component of evaluating the posterior pdf, one can employ model reduction techniques to construct inexpensive-to-solve reduced-order models of the PDEs [8, 24, 34, 40, 49, 65]. On the other hand, one can reduce the cost of evaluating the likelihood directly using polynomial chaos [45, 46]. One can also pose the task of approximating the Bayesian solution as a density estimation problem, for which there is a vast literature, including classical density estimation, multi-dimensional kernel density approximation, and mixture density estimation; see [61, 62] and references therein. Finally, one can reduce the cost of evaluating the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ by approximating this so-called response surface

using such techniques as metamodels or radial basis functions [47, 64], and Gaussian process models [30, 31, 36, 53]. The majority of these methods do not exploit derivative (of the parameter-to-observable map) information, which is our goal here.

Here, we choose to directly approximate the posterior using a Hessian-based Gaussian process response surface. This results in an inexpensive-to-evaluate explicit response surface equipped with an analytical uncertainty estimate. Thus, this "surrogate posterior density" can be sampled, using MCMC for example, at negligible cost compared to sampling the original posterior density. The task of solving a statistical inverse problem therefore reduces to approximating a function over high dimensional parameter space, for which one has to face the curse of dimensionality.

The term "curse of dimensionality" was coined by Bellman [5] in the context of optimization to reflect the fact that in order to obtain an accurate minimizer within ε tolerance, an exponential number of function evaluations, i.e., $\left(\frac{1}{\varepsilon}\right)^N$, is required if our knowledge about the cost function is limited, for example, to Lipschitz continuity. A similar curse of dimensionality in function approximation says that an exponential number of function evaluations, i.e., $\left(\frac{1}{\varepsilon}\right)^N$, is necessary for the approximation to be uniformly accurate within ε tolerance if Lipschitz continuity is all we know about the approximated function [20].

In the context of statistics, the curse of dimensionality reflects the fact that high dimensional spaces are very sparse [61]. For example, the ratio of the volume of the inscribed hypersphere and that of the corresponding hypercube converges to zero as N approaches infinity. Another example is that the volume of a thin shell between hyperspheres of radii r and $r - \epsilon$ converges to the volume of the hypersphere of radius r as N approaches infinity no matter how small ϵ is. These two examples show that the volume content of hypercubes and hyperspheres concentrates near their surfaces. That is, the center of these objects is more or less empty. A concrete example of the sparsity in high dimensional space is the hypercube $[-1, 1]^{10}$ whose first quadrant contains only the fraction 2^{-10} (2^{-N} for N-dimensional space) of uniform sampling data. Furthermore, almost no samples can be found in the inscribed hypersphere.

The problem of approximating a pdf in high dimensions by sample points is a good example of this effect. Since the integral of a bona-fide pdf over the domain of interest is at most unity, the pdf must be negligible everywhere except in the neighborhood of the modes. In addition, if the modes are located away from the boundaries of the domain of interest (which is true for most practical applications in which we choose the domain of interest to be sufficiently large to contain all the important features of the problem under consideration), random sampling methods (especially space-filling techniques) will tend to fail to find the high probability regions, since almost no samples will be in the neighborhood of these regions. In other words, the pdf at the sampling points will most likely be close to zero, and hence any reasonable estimation or interpolation methods based on these values will yield flat response surfaces whose values are close to zero.

Nevertheless, the curse of dimensionality is not entirely a pessimistic result. In fact, it implies that one might be able to reduce its impact if higher order information, for example, gradients and Hessians of the pdf, is exploited. This has indeed been the case for optimization of systems governed by PDEs (i.e., PDE-constrained optimization), where the combination of (Hessian-based) inexact Newton methods with appropriate preconditioners yields methods that can deliver solutions at the cost of a constant number of forward PDE solves, independent of the dimension of the optimization variable space (e.g., [6, 7, 21, 33]). That is, using a suitable class of Newton

methods for optimization and under favourable conditions, the curse of dimensionality in optimization can be mitigated, at least for locating local minima.

A natural idea is therefore to cast the density approximation problem as an optimization problem for which the effect of the curse of dimensionality can be lessened by employing higher order derivatives. In particular, we pose the sampling task (i.e., the task of selecting training points) for the Gaussian process approximation as a sequence of optimization problems (solved by Newton methods) that seek to maximize the error between the Gaussian process approximation and the underlying true pdf. These points also tend to be points of high probability density of the underlying pdf. Moreover, a "piecewise" Gaussian approximation to the underlying pdf is adaptively constructed with local covariance matrices that are inverses of Hessians of the negative log posterior evaluated at the interpolation points. (As is well known, when the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ is linearized, the posterior covariance matrix is equivalent to the inverse of this Hessian.) This proposed Hessian-based Gaussian process method for Bayesian interpolation exploits previous work on adaptive choice of interpolation points in reduced model construction using a greedy algorithm [11, 14, 27].

The remainder of the paper is organized as follows. Section 2 reviews important characteristics of conventional Gaussian process response surfaces, among which are the equivalence with radial basis function approximation, and with Bayesian interpolation. This motivates us to develop a non-stationary Hessian-based Gaussian process in Section 3, followed by a heuristic adaptive sampling strategy for computer experimental design in Section 4. Next, Section 5 provides an analysis of our proposed approach. Section 6 details the choice of the error function, numerical optimization methods, initialization, and how to update the training set. Verification of the proposed response surface methodology is carried out in Section 7 for several probability density approximation problems, including the problem of Bayesian inference of the shape of a scatterer from noisy observations of scattered electromagnetic waves. Finally, Section 8 concludes the paper and discusses some ongoing research issues.

2. Standard Gaussian process response surfaces. We start by reviewing the standard Gaussian process response surface methodology. In order to avoid unnecessary confusions with the Bayesian inversion described above, we rename the posterior density solution of the inverse problem $\pi_{\mathrm{post}}(\mathbf{m})$ as $d(\mathbf{m})$ for which we seek to find an approximation. Assume we are given a training set $\{\mathbf{m}_i, d_i = d(\mathbf{m}_i)\}_{i=1}^n$ where $\mathbf{m}_i \in \mathbb{R}^N$ is a point (training site) in the N-dimensional parameter space and its corresponding function evaluation is d_i. If the training set is noise-free, which is the case in this paper since we simply evaluate $d(\mathbf{m}_i)$, the Gaussian process response surface method is a Bayesian interpolation technique that aims to statistically interpolate the unknown underlying function $d(\mathbf{m})$ given the training set [56]. The main idea behind Gaussian process response surface methods is to assume the unknown deterministic function $d(\mathbf{m})$ to be a random function realization generated from a Gaussian process prior. The Gaussian process prior should be therefore sufficiently flexible, which is assumed from now on, so that ideally there exists a random function realization that is indistinguishable from the unknown function. Once the observable data are obtained, they are combined with the Gaussian process prior through a Bayesian framework to produce predictions for the unknown function $d(\mathbf{m})$.

By definition, a random function $d(\mathbf{m})$ is a Gaussian process if the marginal density $\pi(d(\mathbf{m}_1), d(\mathbf{m}_2), \ldots, d(\mathbf{m}_n))$ is a multivariate Gaussian, for any set of points $\{\mathbf{m}_i\}_{i=1}^n$. A Gaussian process is completely determined by a mean function $\mu(\mathbf{m})$ and

a covariance function $k\left(\mathbf{m}_i, \mathbf{m}_j\right)$ for two arbitrary points $\mathbf{m}_i$ and $\mathbf{m}_j$. Assume that these functions are given for now (their constructions are the subjects of Sections 3.1 and 3.2). By assigning a Gaussian process prior on the random function $d(\mathbf{m})$, the joint distribution of $d(\mathbf{m}^*)$ with n observations $\mathbf{d}_{obs} = [d(\mathbf{m}_1), d(\mathbf{m}_2), \ldots, d(\mathbf{m}_n)]^T$ at n training points $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_n]$ is a Gaussian given by

$$\pi\left(d(\mathbf{m}^*), \mathbf{d}_{obs}|\mathbf{M}, \mathbf{m}^*\right) = \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{\mu}_{obs} \\ \mu(\mathbf{m}^*) \end{array}\right], \left[\begin{array}{cc} K(\mathbf{M}, \mathbf{M}) & K(\mathbf{M}, \mathbf{m}^*) \\ K^T(\mathbf{M}, \mathbf{m}^*) & k(\mathbf{m}^*, \mathbf{m}^*) \end{array}\right]\right), \quad (2.1)$$

where the matrix $K(\mathbf{M}, \mathbf{M})$ is computed as $K_{ij} = k(\mathbf{m}_i, \mathbf{m}_j)$, and $k(\mathbf{m}_i, \mathbf{m}^*)$ the ith element of the column vector $K(\mathbf{M}, \mathbf{m}^*)$. The availability of the training points is assumed for now, and in Section 4 we will present an adaptive sampling method to select these points. Using conditional distribution of multivariate normal [56, 59], the posterior distribution of $d(\mathbf{m}^*)$ is given by

$$\pi_{\text{post}}\left(d(\mathbf{m}^*)|\mathbf{M}, \mathbf{d}_{obs}, \mathbf{m}^*\right) = \mathcal{N}\left(E\left\{d(\mathbf{m}^*)\right\}_{\text{post}}, var\left\{d(\mathbf{m}^*)\right\}_{\text{post}}\right), \quad (2.2)$$

where the expectation and variance read

$$E\left\{d(\mathbf{m}^*)\right\}_{\text{post}} = \underbrace{\mu(\mathbf{m}^*)}_{\mu(\mathbf{m}^*)_{\text{prior}}} + K^T(\mathbf{M}, \mathbf{m}^*)\left[K(\mathbf{M}, \mathbf{M})\right]^{-1}(\mathbf{d}_{obs} - \boldsymbol{\mu}_{obs}), \quad (2.3)$$

$$var\left\{d(\mathbf{m}^*)\right\}_{\text{post}} = \underbrace{k(\mathbf{m}^*, \mathbf{m}^*)}_{var\{d(\mathbf{m}^*)\}_{\text{prior}}} - K^T(\mathbf{M}, \mathbf{m}^*)\left[K(\mathbf{M}, \mathbf{M})\right]^{-1} K(\mathbf{M}, \mathbf{m}^*). \quad (2.4)$$

Since $\mathbf{m}^*$ is arbitrary, equation (2.2) is the posterior distribution of $d(\mathbf{m})$ at any $\mathbf{m}$ in the parameter space. The Bayesian interpretation is now clear as follows. Equation (2.3) states that the posterior mean function at $\mathbf{m}^*$ is the corrected version of the prior mean function using the observation (or measurement) information encoded in the second term on the right side. Furthermore, the posterior error bar (or the posterior uncertainty) is reduced once the prior knowledge and observations are combined as shown in equation (2.4). Indeed, since the covariance matrix $K(\mathbf{M}, \mathbf{M})$ is symmetric positive definite, and hence its inverse, the second term on the right side is positive, which shows that $var\left\{d(\mathbf{m}^*)\right\}_{\text{post}} \leq var\left\{d(\mathbf{m}^*)\right\}_{\text{prior}}$.

We now discuss some other properties of Gaussian process response surfaces that are useful for our subsequent developments. To begin, we introduce the mean squared prediction error (MSPE) with respect to a distribution. Following Santner *et al.* [59] we define

$$MSPE\left(\hat{d}(\mathbf{m}^*), F_n\right) = \mathbb{E}\left\{\left[\hat{d}(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_n}, \quad (2.5)$$

where $\hat{d}(\mathbf{m}^*)$ is a generic predictor of $d(\mathbf{m}^*)$, F_n denotes the joint distribution of $(\mathbf{d}_{obs}, d(\mathbf{m}^*))$, i.e., the distribution in (2.1). The following theorem summarizes some important properties of Gaussian process response surfaces.

THEOREM 2.1. *The following properties hold for the Gaussian process defined in* (2.1)

i) The predicted mean function (2.3) *is the unique minimizer of the MSPE with respect to joint distribution* (2.1)*. Furthermore, it is a linear and unbiased predictor. That is, it is the best linear unbiased predictor (BLUP).*

ii) The predicted mean function (2.3) *interpolates the unknown functions at all points in the training set* **M**.

iii) The mean square prediction error incurred by the predicted mean function is exactly the variance (2.4).

Proof. See Santner *et al.* [59] for a proof. □

The first assertion of Theorem 2.1 therefore suggests that one should use the predicted mean function (2.3) as the predictor for the unknown random function $d(\mathbf{m})$. The second assertion implies that the variance (2.4) is zero at all the training points, that is, the predicted uncertainty at the training points is zero. Hence all the random functions generated by (2.2) interpolate the observed data. Moreover, the third assertion implies that the posterior variance (2.4) can be used as a measure of uncertainty for the Gaussian process predictor (2.3).

We next relate the mean predictor (2.3) with radial basis interpolations. If the covariance function $k(\cdot, \cdot)$ is of radial-basis-function type, the predicted mean function can be shown to be a radial basis interpolation as follows [56]. Let us denote

$$\boldsymbol{\alpha} = [K(\mathbf{M}, \mathbf{M})]^{-1} (\mathbf{d}_{obs} - \boldsymbol{\mu}_{obs}).$$

Substitute $\boldsymbol{\alpha}$ into (2.3) we obtain

$$\hat{d}_n(\mathbf{m}^*) = \mu(\mathbf{m}^*) + \sum_{i=1}^{n} \alpha_i k(\mathbf{m}_i, \mathbf{m}^*), \tag{2.6}$$

which is a radial basis interpolation of the error between the predictor and the prior mean. As a result, the predictor inherits the regularity of the covariance function, assuming the prior mean is sufficiently regular. The importance of the mean function and the covariance function is now clear. They reflect our prior knowledge about what the unknown function $d(\mathbf{m})$ is likely to be. For example, if the underlying function is expected to be not very nonlinear and infinitely smooth, the mean function can be chosen to be linear, and a squared exponential function (also known as Gaussian kernel)

$$k(\mathbf{m}_i, \mathbf{m}_j) = \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{\Sigma}^{-1}}^2\right), \tag{2.7}$$

can be used as the covariance function. Here, we have defined the Mahalanobis norm as

$$\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{\Sigma}^{-1}} = \sqrt{(\mathbf{m}_i - \mathbf{m}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{m}_i - \mathbf{m}_j)}, \tag{2.8}$$

with a positive definite matrix $\boldsymbol{\Sigma}$. Typically, $\boldsymbol{\Sigma}$ is chosen to be a constant diagonal matrix whose diagonal entries are inferred from the training data [42, 48]. Each diagonal entry corresponding to each dimension can be considered as the length scale over which the predictor changes significantly in that particular dimension. Statistically, these length scales determine distance between two points in each dimension such that the predictions at these points are uncorrelated. In other words, these length scales reflect our beliefs about the smoothness of the unknown function $d(\mathbf{m})$. For example, if $\boldsymbol{\Sigma}_{ii}$, the ith diagonal entry, is small, the predictor varies rapidly in the ith dimension while it tends to be constant for large $\boldsymbol{\Sigma}_{ii}$.

3. Non-stationary adaptive Hessian-based Gaussian process response surfaces.

3.1. Prior mean construction. We next discuss on how to choose the prior mean function and postpone the construction of the covariance function to Section 3.2. For our problem of interest where the unknown underlying function $d(\mathbf{m})$ is a probability density, its mean value can be estimated as

$$d_{\text{mean}} = \frac{\int_\Omega d(\mathbf{m})\, d\Omega}{m(\Omega)} \leq \frac{1}{m(\Omega)},$$

where $m(\Omega)$ denotes the measure of Ω, and the inequality is obtained from the fact that the domain of interest Ω is a subset high dimensional parameter spaces over which $d(\mathbf{m})$ is a *bona fide* density, i.e., $\int_{\mathbb{R}^N} d(\mathbf{m})\, d\Omega = 1$. Clearly, the mean value d_{mean} is small if the measure of the domain of interest is sufficiently large. We therefore expect that zero-mean is a good prior information. This is intuitively meaningful since $d(\mathbf{m})$, the Bayesian posterior probability density, is typically significant only in the neighborhood of the modes, while it tends to be small or close to zero elsewhere. On the other hand, since the approximation approaches the prior mean for points that are further away from the training set, the zero-mean prior permits reasonable approximations for regions with small probability density.

3.2. Adaptive non-stationary covariance function. Covariance functions that are a function of only relative distance between two points, e.g., equation (2.7) with constant $\boldsymbol{\Sigma}$, are known as stationary covariance functions. However, Gaussian processes with stationary covariance function can provide accurate predictors only for functions with nearly constant smoothness since stationary lacks the ability to adapt to variable smoothness of the unknown function of interest. A number of non-stationary covariance functions have been devised in literature, see [23, 26, 32, 37, 50, 51, 54–56, 58, 60] for examples. Below, we rationalize the derivation of our Hessian-based non-stationary covariance functions.

We begin by re-examining the predictor (2.6) with zero-mean Gaussian process prior as argued in Section 3.1:

$$\hat{d}_n(\mathbf{m}^*) = \sum_{i=1}^n \alpha_i k(\mathbf{m}_i, \mathbf{m}^*). \tag{3.1}$$

If $n = 1$ and if the covariance is of Gaussian-type as in equation (2.7), then the predictor in (3.1) is nothing more than a Gaussian approximation to $d(\mathbf{m})$ where the covariance matrix is given by $\boldsymbol{\Sigma}$. If, in addition, $\boldsymbol{\Sigma}^{-1}$ is the hessian of $-\ln d(\mathbf{m})$, i.e., $\boldsymbol{\Sigma}^{-1} = \nabla^2(-\ln d(\mathbf{m}))$, then the predictor becomes the popular Laplace approximation (see [43] and references therein). That is, the predictor is exact if the underlying density $d(\mathbf{m})$ is a Gaussian whose peak is at $\mathbf{m}_1$.

For $n > 1$, it is natural and intuitive to generalize the Laplace approximation idea by combining local Laplace approximations constructed in different sub-domains. The challenge is how to combine them to form a global approximation. Our idea is the following. Since the Laplace approximation is locally an accurate approximation, the contribution of $k(\mathbf{m}_i, \mathbf{m}^*)$ to the predictor should dominate the other terms $k(\mathbf{m}_j, \mathbf{m}^*)$ for $j \neq i$, if $\mathbf{m}^*$ is closest to $\mathbf{m}_i$. In order to fulfill this goal, we introduce the following non-stationary covariance function

$$k(\mathbf{m}_i, \mathbf{m}_j) = \sum_{l=1}^L P(z = l|\mathbf{m}_i) P(z = l|\mathbf{m}_j) \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{H}_l}^2\right), \tag{3.2}$$

where $\boldsymbol{H}_l = \nabla^2\left(-\ln d(\mathbf{m}_l)\right)$ and $L \leq n$ (to be shown in Section 4). $P(z = l|\mathbf{m}_i)$ can be considered as the conditional probability of having selected the lth kernel $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right)$ given $\mathbf{m}_i$, and z is known as latent indicator variable. For example, $P(z = l|\mathbf{m}_i)$ should approach 1 if $\mathbf{m}_i \to \mathbf{m}_l$, and zero if $\mathbf{m}_i$ is far away from $\mathbf{m}_l$. In particular, as derived in Section 3.3, the following form of $P(z = l|\mathbf{m}_i)$ satisfies the requirements

$$P(z = l|\mathbf{m}_i) = \frac{\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_l\|^2_{\boldsymbol{H}_l}\right)}{\sum_{p=1}^{L} \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_p\|^2_{\boldsymbol{H}_p}\right)}, \tag{3.3}$$

which can be seen as a special form of a well-known class of *soft-max* gating networks in machine learning community [9]. Note that the denominator is just a normalized constant while the numerator is a Gaussian with mean $\mathbf{m}_l$ and inverse covariance matrix $\boldsymbol{H}_l$.

The importance of the Hessian information of the underlying posterior $d(\mathbf{m})$ is now explained. It can be seen that the Hessians appear two times in the covariance function definition (3.2). First, the appearance in the kernels $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right)$ ensures that the predictor is the desired piecewise Laplace approximation. Second, the role of the Hessians in the membership probability $P(z = l|\mathbf{m}_i)$ is to guide the co-variance function to pick the appropriate dominant kernel in the predictor. Note that the product $P(z = l|\mathbf{m}_i)P(z = l|\mathbf{m}_j)$ is necessary because it not only guarantees the symmetry of the covariance function but also determines the appropriate covariance structure and hence the smoothness of the predictor. By definition, $k(\mathbf{m}_i, \mathbf{m}_j)$ is the covariance between $d(\mathbf{m}_i)$ and $d(\mathbf{m}_j)$. The covariance in turn encodes the smooth-ness of the predictor. Moreover, the smoothness of a surface is typically measured by its second derivatives, i.e., the Hessian. These suggest that the Hessians should be used in the covariance to shape the smoothness of the predictor accordingly. Our covariance function in (3.2) is built based on this intuition (and on the desire to have a piecewise Laplace approximation). That is, if $\mathbf{m}_i$ and $\mathbf{m}_j$ are close to $\mathbf{m}_l$, with respect to the norm $\|\cdot\|_{\boldsymbol{H}_l}$, the lth term of the sum on the right hand side of (3.2) dominates the covariance while its contribution to the covariance is small if either $\mathbf{m}_i$ or $\mathbf{m}_j$ is far away from $\mathbf{m}_l$.

Under the piecewise Laplace approximation view point, the membership prob-ability is served as an automatic mechanism to partition the high dimensional pa-rameter space into overlapping sub-regions over which the posterior density of the inverse problem is dominantly interpolated and approximated by local Gaussians. As a demonstration, Figure 3.1 shows three membership probabilities corresponding to three modes of the exact density $d(\mathbf{m}) = \frac{1}{3}\mathcal{N}(-4, 1) + \frac{1}{3}\mathcal{N}(0, 0.75) + \frac{1}{3}\mathcal{N}(4, 0.5)$. As can be seen, each membership probability is one if $\mathbf{m}_i$ is close to its corresponding mode and zero otherwise. Sub-regions dominated by the first and the second mem-bership probabilities do not overlap while they do with that of the third one. This reflects the fact the first and second modes do not overlap, but they do with the third one. The role of the membership probabilities in automatically identifying the dominant local Laplace approximation is clearly demonstrated in this figure.

A question that needs to be addressed is whether (3.2) defines a valid covariance function. This is important since a Gaussian process exists if and only if the covariance function is valid, according to the Kolmogorov's existence theorem [2]. The following result answers this question.

THEOREM 3.1. *Assume* $L \geq 1$ *and* $\boldsymbol{H}_l$ *is symmetric positive definite* $\forall l =$
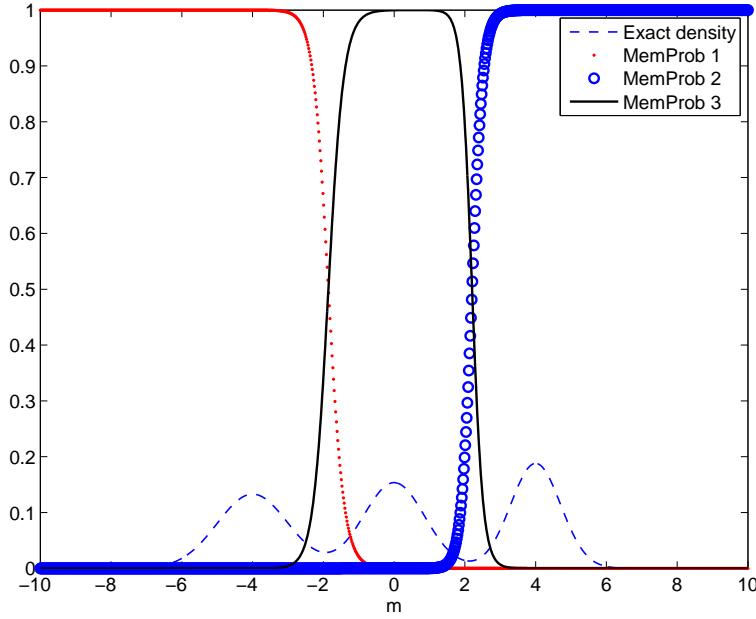
FIG. 3.1. *Three membership probabilities corresponding to three modes of the exact density* $d(\mathbf{m}) = \frac{1}{3}\mathcal{N}(-4,1) + \frac{1}{3}\mathcal{N}(0,0.75) + \frac{1}{3}\mathcal{N}(4,0.5)$.

$1, \ldots, L$. *Then*

$$k(\mathbf{m}_i, \mathbf{m}_j) = \sum_{l=1}^{L} P(z=l|\mathbf{m}_i)P(z=l|\mathbf{m}_j)\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{H}_l}^2\right)$$

is a valid non-stationary covariance function.

Proof. The non-stationary is clear since $k(\mathbf{m}_i, \mathbf{m}_j)$ a function of not only the relative distance between $\mathbf{m}_i$ and $\mathbf{m}_j$ but also $\mathbf{m}_i$ and $\mathbf{m}_j$. The symmetry is also clear. In order to prove that the kernel $k(\mathbf{m}_i, \mathbf{m}_j)$ is positive definite, it is sufficient to show that the matrix $\mathbf{K}(\mathbf{M}, \mathbf{M})$ is positive definite $\forall n \geq 1$. We begin by the following fact

$$\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{H}_l}^2\right) = \frac{1}{(\pi/2)^{N/2}|\boldsymbol{H}_l^{-1}|^{1/2}} \times$$
$$\int_{\mathbb{R}^N} \exp\left(-\|\mathbf{m}_i - \mathbf{m}\|_{\boldsymbol{H}_l}^2\right)\exp\left(-\|\mathbf{m}_j - \mathbf{m}\|_{\boldsymbol{H}_l}^2\right) d\Omega.$$

Thus, $\forall n \geq 1, \forall \mathbf{c} \in \mathbb{R}^n$, we have

$$\mathbf{c}^T\mathbf{K}(\mathbf{M}, \mathbf{M})\mathbf{c} = \sum_i\sum_j c_ic_jk(\mathbf{m}_i, \mathbf{m}_j) = \sum_l \frac{1}{(\pi/2)^{N/2}|\boldsymbol{H}_l^{-1}|^{1/2}} \times$$
$$\int_{\mathbb{R}^N}\left\{\sum_i c_iP(z=l|\mathbf{m}_i)\exp\left(-\|\mathbf{m}_i - \mathbf{m}\|_{\boldsymbol{H}_l}^2\right)\right\}^2 d\Omega \geq 0.$$

It is clear that the equality happens if and only if the term in the curly bracket is zero almost everywhere which in turn happens if and only if $c_i = 0, \forall i = 1, \ldots, n$, and this completes the proof. $\square$

The importance of Theorem 3.1 is now clear. It ensures the positive definiteness of $\mathbf{K}(\mathbf{M}, \mathbf{M})$, which in turn guarantees its invertibility that is necessary for (2.3) and (2.4).

3.3. Derivation of the membership probabilities. Recall that the membership probability $P(z = l | \mathbf{m}_i)$ is the conditional probability of having selected the lth kernel $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right)$ given $\mathbf{m}_i$. Using the Bayes' theorem gives

$$P(z = l | \mathbf{m}_i) = \frac{P(\mathbf{m}_i | z = l) \times P(z = l)}{\sum_{p=1}^{L} P(\mathbf{m}_i | z = p) \times P(z = p)},$$

where $P(\mathbf{m}_i | z = l)$ and $P(z = l)$ are the likelihood and prior of selecting the lth kernel, respectively. Since our prior knowledge is vague, we choose $P(z = l) = 1/L, \forall l = 1, \ldots, L$. The likelihood is chosen to be the (unnormalized) Gaussian $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_l\|^2_{\boldsymbol{H}_l}\right)$ which reflects our desire that the likelihood must be large, in the Mahalanobis norm $\|\cdot\|_{\boldsymbol{H}_l}$, as $\mathbf{m}_i$ approaches the mean $\mathbf{m}_l$. The extent to which the likelihood is still significant is determined by the curvature of the true unknown $d(\mathbf{m})$, which appears as the inverse of the likelihood covariance matrix. Hence, the final form of the membership probability is given as in equation (3.3).

4. Sequential adaptive sampling strategy. This section addresses the computer experimental design issue on how to look for the training points adaptively. The method we are going to describe follows our previous work on scalable adaptive algorithms for constructing reduced models in high-dimensional parameter spaces [11,14]. To begin, we define the following generic error function

$$\mathcal{G}(\mathbf{m}, n), \tag{4.1}$$

which is a function of parameters $\mathbf{m}$ and the training set size n (and the training set itself, but is omitted here for simplicity). The error could be, for example, the squared error between the true function $d(\mathbf{m})$ and the predictor $\hat{d}_n(\mathbf{m})$

$$\mathcal{G}(\mathbf{m}, n) = \left[\hat{d}_n(\mathbf{m}) - d(\mathbf{m})\right]^2, \tag{4.2}$$

or the predictive variance in (2.4) as the error indicator. Generally, for Algorithm 1 and its corresponding theory to work, it is desirable that the cost $\mathcal{G}(\mathbf{m}, n)$ be less then some small tolerance ε at all training points. The squared error and the predictive variance clearly satisfy this requirement since they are zero at all the training points. We first outline the adaptive sampling algorithm as follows.

ALGORITHM 1.

Adaptive Sampling Algorithm

1. *Given a set of training points $\{\mathbf{m}_i, d_i = d(\mathbf{m}_i)\}_{i=1}^{n}$, solve the optimization problem*

$$\max_{\mathbf{m} \in \Omega} \mathcal{G}(\mathbf{m}, n), \tag{4.3}$$

   to find the location in parameter space at which the error is maximized, i.e. find $\mathbf{m}^* = \arg\max \mathcal{G}(\mathbf{m}, n)$.
2. *If $\mathcal{G}(\mathbf{m}^*, n) < \varepsilon$, where ε is the desired level of accuracy, then terminate the algorithm. If not, go to the next step.*
3. *With $\mathbf{m} = \mathbf{m}^*$, compute the true function $d(\mathbf{m}^*)$. Update the predictor. Go to Step 1.*

The first step of the adaptive sampling algorithm incrementally find the next training points at locations where $\mathcal{G}(\mathbf{m}, n)$, as a measure of the error between the underlying true posterior density and the current Gaussian process response surface approximation, is maximized. The structure of the optimization problem that must be solved in each adaptive cycle is similar to that of PDE-constrained inverse problems, and hence, many of the associated tools for large-scale optimization can be recruited, in particular Newton-CG solvers, trust-region globalization, and Eisenstat-Walker inexactness, e.g., [3, 4, 6, 7, 11, 21]. In order to make Algorithm 1 well-defined, as discussed in the next section, the initial guess is admissible if the error function is at least ε.

If the maximum error is less than the prescribed tolerance ε in Step 2, the algorithm stops. Otherwise, Step 3 will update the current predictor and return to Step 1. In particular, the training set is updated using the maximizer $\mathbf{m}^*$ and its corresponding true posterior value $d(\mathbf{m}^*)$. If the Hessian of the negative log posterior, $\nabla^2 \left(-\ln d(\mathbf{m}^*) \right)$, is positive definite, the adaptive covariance function (3.2) and the membership probability (3.3) are updated by increasing L by one. That is, we build a local Gaussian approximation whose inverse covariance is the local Hessian of the negative log posterior. This local Gaussian approximation is then used to update the Gaussian process covariance function and the membership probability. Effectively, the method builds an adaptive sparse non-stationary Gaussian process that is generally improved after each cycle. The method identifies regions in high dimensional spaces where the discrepancy of the response surface is maximal, and then inserts local Gaussian approximations at those points to drive the response surface error down. Since the Gaussian process response surface is interpolating, the resulting approximation is identical to the posterior density at these points. Furthermore, the Hessian ensures that the response surface locally adapts to the shape of posterior $d(\mathbf{m})$ accurately.

5. An analysis of the adaptive Gaussian process response surface. In this section, each adaptive cycle in Algorithm 1 is analyzed to show that the whole algorithm is well defined. We first show that the optimization problem (4.3) has a solution under suitable assumptions.

PROPOSITION 5.1. *Assume $\mathcal{G}(\mathbf{m}, n)$ is a continuous function of $\mathbf{m} \in \Omega$, where Ω is closed and bounded subset of $\mathbb{R}^N$. Then there exists a solution for the optimization problem (4.3).*

Proof. See [11, 14] for a proof. □

The closedness and boundedness of the domain of interest Ω are reasonable. For example, in the shape inverse electromagnetic scattering problem studied in Section 7, the shape parameters $\mathbf{m}$ are bounded due to our prior belief in the boundedness of the shape. Proposition 5.1 therefore implies that Steps 2 and 3 of Algorithm 1 always go through, thus each cycle certainly finishes.

Revisiting previous sampled points is an expensive task requiring forward and adjoint solves, and hence should be avoided. On the other hand, distinction of sampled points implies the non-singularity of $\mathbf{K}(\mathbf{M}, \mathbf{M})$, which is vital in ensuring the existence and uniqueness of the the predictor and its uncertainty in (2.3)–(2.4). Our next result shows that in fact Algorithm 1 always finds new sampling points.

THEOREM 5.2. *Algorithm 1 is well-defined in the sense that (i) it terminates in finite time and (ii) all sampled points are distinct.*

Proof. See [11, 14] for a proof. □

The next question needs to be resolved is whether the proposed adaptive training (also known as active learning) can systematically bias the inference. Since Bayesian

inference is consistent with the *likelihood principle* [41] which states that the inference should depend only on the likelihood of the data that is actually observed, one is free to choose training points without introducing any bias to the inference.

We next show that our piecewise Laplace (Gaussian) approximation (2.3) improves as the number of training points increases.

THEOREM 5.3. *Denote $\hat{d}_n(\mathbf{m}^*)$ as the mean predictor* (2.3) *and assume $\hat{d}_{n+1}(\mathbf{m}^*) \neq \hat{d}_n(\mathbf{m}^*), \forall n \in \mathbb{N}$. As the number of training points increases, the mean squared prediction error decreases in the following sense:*

$$\mathbb{E}\left\{\left[\hat{d}_{n+1}(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_{n+1}} < \mathbb{E}\left\{\left[\hat{d}_n(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_n}.$$

Proof. We have the following inequalities for the MSPE

$$\mathbb{E}\left\{\left[\hat{d}_{n+1}(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_{n+1}} \leq \mathbb{E}\left\{\left[\hat{d}(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_{n+1}}$$

$$< \mathbb{E}\left\{\left[\hat{d}_n(\mathbf{m}^*) - d(\mathbf{m}^*)\right]^2\right\}_{F_n},$$

where $\hat{d}(\mathbf{m}^*)$ denotes an arbitrary linear predictor. The first inequality holds true due to the minimization property of $\hat{d}_{n+1}(\mathbf{m}^*)$ in the first part of Theorem 2.1. The second inequality follows from choosing $\hat{d}(\mathbf{m}^*) = \hat{d}_n(\mathbf{m}^*)$ and applying the marginalization property of the multivariate Gaussian (2.1). Note that the second inequality is strict due to the assumption $\hat{d}_{n+1}(\mathbf{m}^*) \neq \hat{d}_n(\mathbf{m}^*)$ and the uniqueness of the minimizer $\hat{d}_{n+1}(\mathbf{m}^*)$.
□

6. Error function, numerical optimization, initialization, and training set update. The active learning method proposed in Section 5 works for a class of quite general error functions. For our purpose, the true error $\left[\hat{d}_n(\mathbf{m}) - d(\mathbf{m})\right]^2$ turns out to be a good candidate, as we now explain. Recall that the main goal of this paper is to find as many modes as possible and then to interpolate the expensive-to-evaluate posterior density function $d(\mathbf{m})$ using a piecewise Laplace approximation. Intuitively, the interpolation is statistically more accurate if it captures most of significant probability regions of $d(\mathbf{m})$. In order to fulfill this goal heuristically, we design Algorithm 1 to place training points where the discrepancy between the predictor and true function is largest (at least locally). Due to the local Gaussian nature of the predictor, if $\mathbf{m}$ is sufficiently far away from $\mathbf{m}_l, \forall l = 1, \ldots, L$, the predictor $\hat{d}_n(\mathbf{m})$ will approach the prior mean, which is zero, and hence the true error $\left[\hat{d}_n(\mathbf{m}) - d(\mathbf{m})\right]^2$ will approach $[d(\mathbf{m})]^2$. As a result, the worst-case scenario error found in each adaptive cycle is most likely a mode of $d(\mathbf{m})$.

As also discussed in Section 5, Ω is typically generated by simple bound constraints on parameters $\mathbf{m}$. Similar to our previous work [11, 14], we choose to solve the bound-constrained optimization:

$$\max_{\mathbf{m}} \mathcal{G}(\mathbf{m}), \tag{6.1}$$

subject to

$$\mathbf{m}_{min} \leq \mathbf{m} \leq \mathbf{m}_{max}, \tag{6.2}$$

using a subspace trust region interior reflective inexact Newton conjugate gradient method described in [11].

Initialization is one of the important factors determining the cost that the optimization solver takes to converge. In particular, if the initial guess is far away from the optimizer, it might take several iterations for the optimization solver to move to the basin of attraction where the designed convergence rate takes place, e.g., quadratic, if a Newton method is employed. Therefore in order to reduce the cost, it is vital to find a good initial guess for the optimization problem in each greedy cycle. Clearly, the simplest way is to take a random initialization which is most likely not to be close to the basin of attraction of any local optimizers. Since the MSPE (2.4) is analytical and cheap to evaluate, another straightforward idea is to find the point where the MSPE is maximized and then use it as the initial guess. However, as noticed by MacKay [41], maximizers of MSPE tend to be at the boundary of the domain Ω, which is not of our interest.

In the context of active learning (adaptive sampling) for Gaussian process, Seo *et al.* [63] numerically shows that using the selection criteria by Cohn [18] yields a more accurate predictor than that proposed by MacKay [41]. The reason is that the Cohn criteria for sampling aims to minimize the mean squared error, and in particular, it maximizes the average reduction in predictive variance. Nevertheless, adaptive sampling using Cohn criteria is expensive as discussed in Christen and Sanso [17]. They propose a cheap alternative, an approximation to the Cohn criteria, over a test set $\mathbf{M}^a = [\mathbf{m}_1^a, \ldots, \mathbf{m}_{n^a}^a]$ of (random or grid) points in the parameter space. Specifically, the selection process is based on the solution of the following optimization problem:

$$\max_{\mathbf{m}_i^a \in \mathbf{M}^a} \mathcal{J}(\mathbf{m}_i^a), \tag{6.3}$$

where

$$\mathcal{J}(\mathbf{m}_i^a) = \frac{\frac{1}{n^a} \sum_{j=1}^{n^a} k(\mathbf{m}_j^a, \mathbf{m}_i^a)^2 + \frac{1}{n^a} I_{\mathbf{M}}^1}{I_{\mathbf{M}}^2 + \sqrt{\sum_{j=1}^n k(\mathbf{m}_i^a, \mathbf{m}_j)^2}},$$

with $\mathbf{m}_j \in \mathbf{M}$ and $\mathbf{m}_j^a \in \mathbf{M}^a$; $I_{\mathbf{M}}^1$ and $I_{\mathbf{M}}^2$ are two constants independent of $\mathbf{m}_i^a$ and are defined as

$$I_{\mathbf{M}}^1 = \sum_{i=1}^{n^a} \sum_{j=1}^n k(\mathbf{m}_i^a, \mathbf{m}_j)^2, \quad I_{\mathbf{M}}^2 = \max_{\mathbf{m}_i \in \mathbf{M}} \sum_{j=1}^n |k(\mathbf{m}_i, \mathbf{m}_j)|.$$

Due to the numerator, the maximizer of the cost in (6.3) should have high predictive variance and be highly correlated with other points in $\mathbf{M}^a$. At the same time, it should not be so close, and hence less correlated, to the current training set $\mathbf{M}$, due to the denominator. Meanwhile, an approach for near-optimal training points has been proposed [39]. However, one has to solve a combinatorial optimization, which we try to avoid here since it is prone to the curse of dimensionality. After all, we need a cheap and reasonably good initial guess and then devote our effort to the continuous optimization problem (4.3) using efficient and scalable optimization solver, which is designed to be immune to the curse of dimensionality. Keeping this goal in mind, we choose the solution of (6.3) as the initial guess for three reasons besides its fast evaluation. First, it is not so close to the current training set about which we have

already learned. Second, since it has high predictive variance it could belong to some region in parameter space where the error may be large. Third, due to its high correlation to other points in $\mathbf{M}^a$, learning this point could provide us information about other unvisited points as well.

The question we would like to address next is how to update the training set $\mathbf{M}$. As discussed in Section 4, the covariance function and the membership probabilities may not be updated after a greedy cycle, depending on whether the Hessian of the negative log posterior $\nabla^2 \left(-\ln d(\mathbf{m}^*) \right)$ is (semi-) positive definite or not. In contrast, the training set $\mathbf{M}$ is always updated after each greedy cycle. One could simply add the maximizer of the optimization problem (4.3) where we locally observe the largest error. This point is currently and intuitively the best to learn about the unknown function $d(\mathbf{m})$. However, in addition to the maximizer, the numerical optimization solver also supplies a whole trajectory of points starting from the initial guess to the optimizer where the unknown function $d(\mathbf{m})$ is evaluated. These points therefore contain information about $d(\mathbf{m})$ about which we are trying to learn. This suggests that we should add the whole trajectory to the current training set. The trade-off is that the condition number of the covariance matrix $K\left(\mathbf{M}, \mathbf{M}\right)$ may increase, and methods for inverting the covariance matrix accurately and efficiently have been addressed elsewhere [25]. Here, we simply use the Cholesky decomposition.

7. Numerical experiments. In this section, the proposed Hessian-based GP predictor is compared to the state-of-the-art radial basis function (RBF) interpolation [10]. Since the standard stationary GP predictor can be shown to be equivalent to a RBF whose kernel function is the same as the GP covariance function [56], we just need to compare our method to a standard GP predictor. To be fair, we also adapt the shape parameter of the radial basis function using the maximum likelihood. To ensure that the cost of generating our GP predictor and the cost of generating the RBF predictor are more or less the same, the number of function evaluations are forced to be the same for both methods. In particular, the number of function evaluations for the Hessian-based GP predictor is counted as the following. First, each forward solve is counted as one. Second, each gradient computation which requires an adjoint solve is counted as one assuming the costs of solving the forward and the adjoint are the same. Third, each Hessian-vector product required in the conjugate gradient iterations is counted as two since a forward-like and an adjoint-like equations have to be solved. Thus to be fair, if the Hessian-based GP uses n_F functions evaluations, the RBF will have n_F interpolation points. As a popular choice, the Latin Hypercube (LHC) sampling is used to generate interpolation points for the RBF approach. Finally, to assess the quality of the predictors, we use several popular discrete norms, namely, the mean squared error, the ℓ^1-norm, the ℓ^2-norm, and the Hellinger norm [16].

7.1. One-dimensional examples. The first example considered in this subsection is a mixture of three Gaussians given by

$$d(\mathbf{m}) = \frac{1}{3}\mathcal{N}\left(-4, 1\right) + \frac{1}{3}\mathcal{N}\left(0, 0.75\right) + \frac{1}{3}\mathcal{N}\left(4, 0.5\right).$$

Figure 7.1 shows the GP predictor together with its uncertainty given by 95% cred-ibility envelope. The quality of the GP predictor (3 greedy cycles) and the RBF predictor, both with 143 function counts, is shown in Table 7.1. As expected, the RBF predictor is more accurate than the GP predictor for low dimensional problem and simple density $d\left(\mathbf{m}\right)$. Note that since the squared error (4.2) is used as the cost

in the adaptive sampling algorithm, the mean squared error tends to be smaller than other measures.
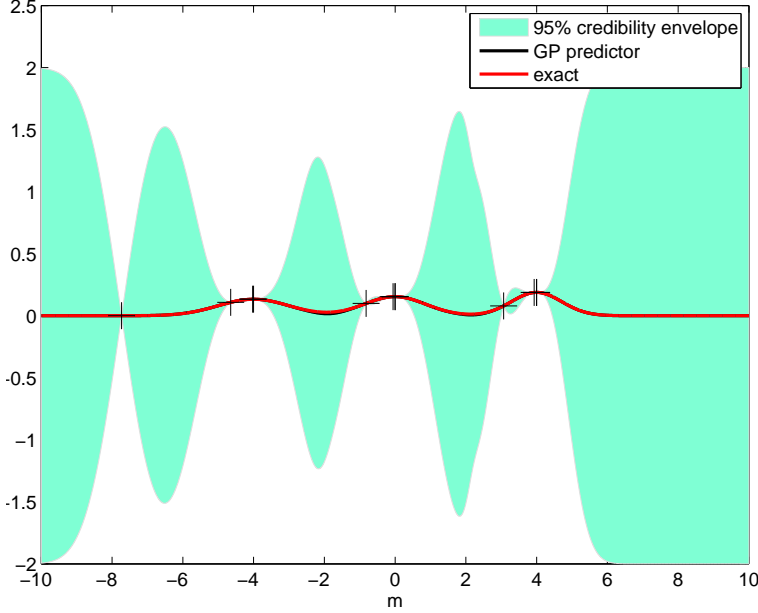

FIG. 7.1. *The GP predictor together with its uncertainty given by 95% credibility envelope versus the exact. Crossed are the points in the training set* **M** *found by the greedy algorithm.*

TABLE 7.1
GP predictor error versus RBF predictor error over a LHC grid of 20,000 points.

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	6.55e-06	4.35	3.62e-01	1.74
RBF	3.77e-16	1.23e-02	2.75e-06	5.31e-03

For more complicated 1D density with anisotropy and localized features such as the one in Figure 7.2, where the true density is given by

$$d(\mathbf{m}) = \sum_{l=0}^{5} (2^{5-l}/63) \mathcal{N} \left(\left[65 - \frac{96}{2^l} \right] / 21, \left(\frac{32}{63} \right)^2 / 2^{2l} \right),$$

the GP predictor (with 371 function counts after 7 greedy cycles) starts to be competitive and this can be seen in Table 7.2.

TABLE 7.2
GP predictor error versus RBF predictor error over a LHC grid of 20,000 points.

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	1.45e-06	3.15	1.70e-01	1.22
RBF	2.55e-3	2.31e1	7.14	1.04e1

As mentioned in Section 6, the ability of our proposed method in seeking the modes of the underlying density $d(\mathbf{m})$ can be observed in Figures 7.1 and 7.2. Again,
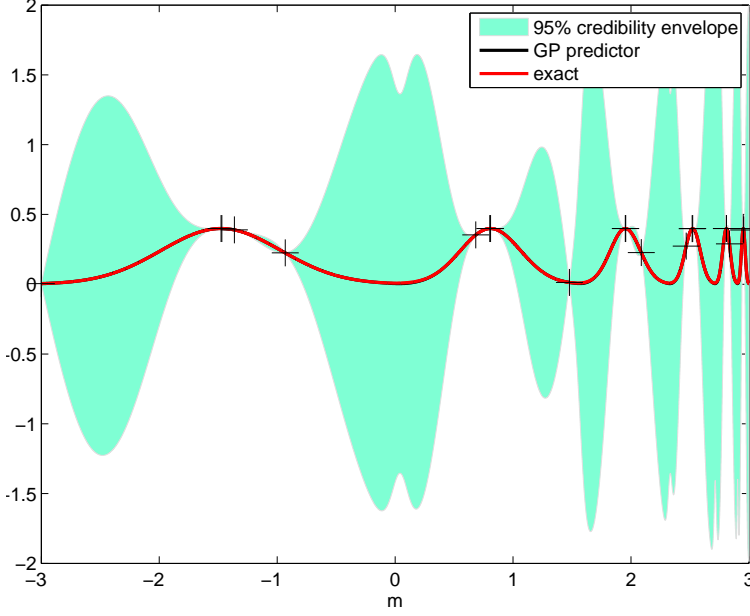
FIG. 7.2.   *The GP predictor together with its uncertainty given by* 95% *credibility envelope versus the exact. Crossed are the points in the training set* **M** *found by the greedy algorithm.*

this is important for our purpose that high probability density regions should be captured as much as possible.

7.2. Two-dimensional example. We consider the following mixture of four anisotropic Gaussian

$$d(\mathbf{m}) \propto \sum_{i=1}^{4} c_i \exp\left(-\frac{1}{2}[\mathbf{m} - \mathbf{m}_i^o]^T H_i^o[\mathbf{m} - \mathbf{m}_i^o]\right),$$

where the local means are given by

$$\mathbf{m}_1^o = [-1.5, -1.5]^T, \mathbf{m}_2^o = [1.5, 1.5]^T, \mathbf{m}_3^o = [-2, 2]^T; \mathbf{m}_4^o = [5, -5]^T,$$

the local inverse covariance matrices read

$$H_1^o = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, H_2^o = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, H_3^o = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}, H_4^o = \begin{bmatrix} 2 & -1.5 \\ -1.5 & 2 \end{bmatrix},$$

and the coefficients c_i are randomly generated as

$$c_1 = 0.25, \quad c_2 = 0.3, \quad c_3 = 0.38, \quad c_4 = 0.07.$$

A GP predictor with 553 function counts after 5 greedy cycles is presented in Figure 7.3(b), whereas the true function is shown in Figure 7.3(a). An adaptive RBF with 553 LHC sampling points is shown in Figure 7.3(c). As can be seen, the GP predictor outperforms the adaptive RBF. To further confirm this, we compute different discrete error norms over 90,000 uniform grid points and present the results in Table 7.3. The errors of the GP predictor are orders of magnitude smaller than those of the adaptive RBF.
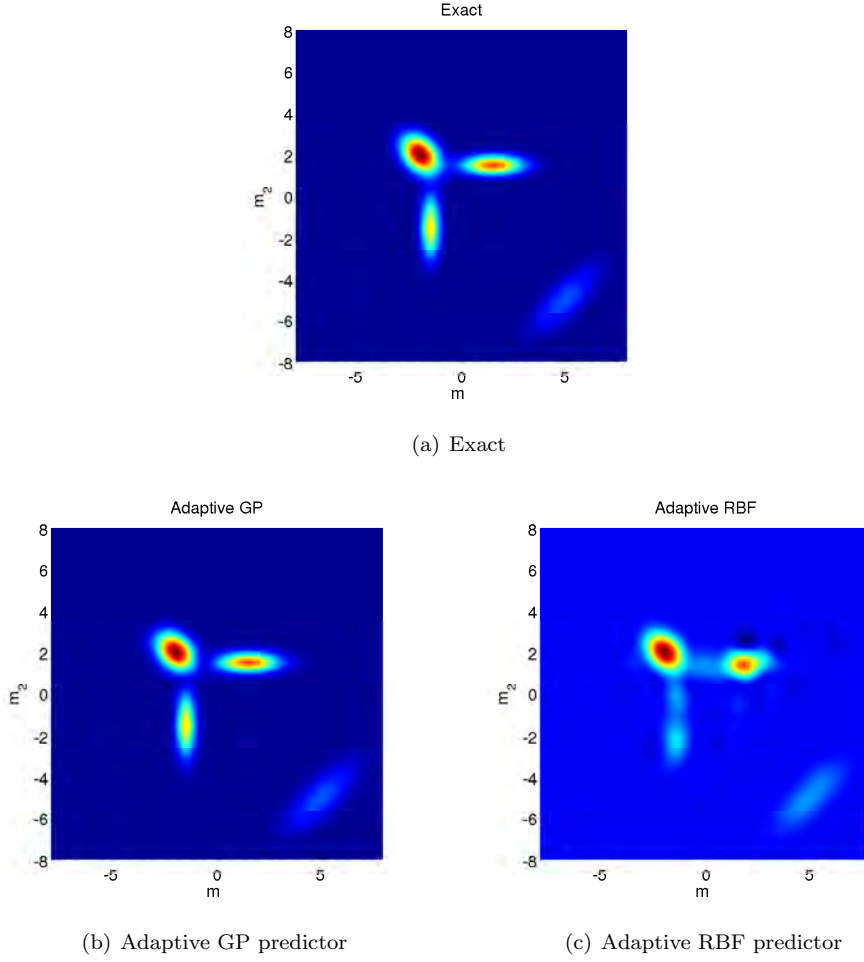
(a) Exact

(b) Adaptive GP predictor

(c) Adaptive RBF predictor

FIG. 7.3.   *A mixture of four anisotropic Gaussians in 2D example; 7.3(a) the exact function, 7.3(b) a GP predictor with 553 function counts after 5 greedy cycles, and 7.3(c) an adaptive RBF predictor with 553 LHC sampling points.*

7.3. Ten-dimensional examples. The first example in this subsection is the mixture of two Gaussians in 10-dimensional space:

$$d(\mathbf{m}) = c_1 \mathcal{N}\left(\mathbf{m}_1^o, \boldsymbol{\Sigma}_1^o\right) + c_2 \mathcal{N}\left(\mathbf{m}_2^o, \boldsymbol{\Sigma}_2^o\right),$$

TABLE 7.3
GP predictor error versus RBF predictor error over a uniform grid of 90,000 points.

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	3.10e-6	4.91	5.28e-1	2.19
RBF	1.36e-4	1.52e1	3.5	9.23

where

$$c_1 = 0.8, c_2 = 0.2,$$
$$m_1 = -2 \times ones(10, 1), m_2 = 2 \times ones(10, 1),$$
$$\mathbf{\Sigma}_1^o = diag[0.1629, 0.1812, 0.0254, 0.1827, 0.1265, 0.0195, 0.0557,$$
$$0.1094, 0.1915, 0.1930],$$
$$\mathbf{\Sigma}_2^o = diag[0.0315, 0.1941, 0.1914, 0.0971, 0.1601, 0.0284, 0.0844,$$
$$0.1831, 0.1584, 0.1919],$$

where $ones(10, 1)$ is a 10×1 column vector with all elements equal to 1, and $diag$ puts a vector on the main diagonal the zero matrix of corresponding size. The domain of interest is the hypercube $[-3, 3]^{10}$. The GP predictor requires three greedy cycles with 679 function evaluations to capture the two modes. In order to compare the exact function $d(\mathbf{m})$ and its adaptive GP predictor, we sample both of them using DRAM [28]; an efficient MCMC toolbox. Table 7.4 shows the sample means from one million MCMC simulations. Here, it is not our attempt to run enough simulations until the MCMC converges, but to show how well the response surface emulates the exact one. As can be seen, the sample means after one million MCMC simulations are the same up to three digits, though they are by no means close to the exact mean $-1.2 \times ones(10, 1)$.

TABLE 7.4
The sample means of GP predictor and the exact function up to one million MCMC simulations.

mean	GP predictor	exact $d(\mathbf{m})$
m_1	-1.9941	-1.9923
m_2	-1.9894	-1.9923
m_3	-2.0007	-1.9996
m_4	-1.9880	-1.9880
m_5	-1.9961	-1.9957
m_6	-1.9995	-1.9997
m_7	-2.0005	-2.0018
m_8	-2.0003	-2.0006
m_9	-1.9911	-1.9847
m_{10}	-1.9891	-1.9902

Now, if we take 679 LHC points for the RBF approach, all the function values evaluated at these points are machine zero, and hence the RBF method would give a zero response surface, which is by no means close to the exact function. This is, as discussed, a manifestation of the curse of dimensionality.

Similarly, we consider the mixture of two 10-dimensional Cauchy distributions:

$$d(\mathbf{m}) = c_1 \mathcal{C}\left(\mathbf{m}_1^o, \boldsymbol{\sigma}_1^o\right) + c_2 \mathcal{C}\left(\mathbf{m}_2^o, \boldsymbol{\sigma}_2^o\right),$$

where

$$\mathcal{C}\left(\mathbf{m}_j^o, \boldsymbol{\sigma}_j^o\right) = \prod_{i=1}^{10} \frac{\sigma_{ji}^o}{\pi \left[\left(\sigma_{ji}^o\right)^2 + \left(\mathbf{m}_i - \mathbf{m}_{ji}^o\right)^2\right]},$$

$$c_1 = 0.65, c_2 = 0.35,$$

$$m_1 = zeros(10,1), m_2 = 2 \times ones(10,1),$$

$$\boldsymbol{\sigma}_1^o = [0.4074, 0.4529, 0.0635, 0.4567, 0.3162, 0.0488, 0.1392,$$
$$0.2734, 0.4788, 0.4824],$$

$$\boldsymbol{\sigma}_2^o = [0.0315, 0.1941, 0.1914, 0.0971, 0.1601, 0.0284, 0.0844,$$
$$0.1831, 0.1584, 0.1919],$$

where $zeros(10,1)$ is the zero vector of dimension 10×1. Similar to the Gaussian case, the domain of interest is the hypercube $[-3,3]^{10}$. The GP predictor requires three greedy cycles with 1386 function evaluations to capture the two modes. Table 7.5 compares the sample means obtained from the GP predictor and the exact $d(\mathbf{m})$ using one million MCMC simulations. The result is reasonable though it is not as good as the Gaussian case. Similar to the Gaussian case, if we use the adaptive RBF method with 1386 LHC points, $d(\mathbf{m})$ is machine zero at these points, and hence yielding a zero response surface! In fact, for both Gaussian and Cauchy cases, we have tested that $d(\mathbf{m})$ is close to machine zero even for 100,000 LHC points. Again, the curse of dimensionality is in action. This observation also suggests that all the discrete norms that we have used above for low dimension examples be useless in high dimensional problems because they are most likely zero.

TABLE 7.5
The sample means of GP predictor and the exact function from one million MCMC simulations.

mean	GP predictor	exact $d(\mathbf{m})$
m_1	2.0002	1.9761
m_2	2.0007	1.8883
m_3	1.9305	1.8976
m_4	1.9980	1.9321
m_5	1.9996	1.9424
m_6	1.9991	1.9962
m_7	1.9970	1.9428
m_8	1.9984	1.9161
m_9	1.9979	1.9236
m_{10}	2.0002	1.8985

7.4. Inverse shape electromagnetic scattering example. In this section, we consider two dimensional transverse magnetic (TM) polarization in the context of electromagnetic scattering due to a scatterer in the free space. The governing

equations read

$$\frac{\partial H_x}{\partial t} + \frac{\partial E_z}{\partial y} = 0 \qquad \text{in } \Omega \times (0, T),$$

$$-\frac{\partial H_y}{\partial t} + \frac{\partial E_z}{\partial x} = 0 \qquad \text{in } \Omega \times (0, T),$$

$$E_z = E_z^I \qquad \text{in } \partial \Omega_S \times (0, T),$$

$$H_x = H_y = E_z = 0 \qquad \text{in } \Omega \times \{0\},$$

where H_x, H_y and E_z denotes the $x - y$ components of the magnetic field and z component of the electric field with appropriate normalization [29], respectively. Here, $E^I = \cos(8(t - x))$ is the incident electric field, and Ω_S the scatterer satisfying $\Omega_s \subset \Omega \subseteq \mathbb{R}^2$.

Next denoting

$$\mathbf{H}_\perp = \begin{bmatrix} -H_y \\ H_x \end{bmatrix}, \quad E = E_z,$$

and using the perfect matched layer (PML) proposed in [1], the above TM equations become

$$\frac{\partial \mathbf{H}_\perp}{\partial t} + \nabla E = \mathbf{L} \qquad \text{in } \Omega \times (0, T), \qquad (7.1a)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \mathbf{H}_\perp = M \qquad \text{in } \Omega \times (0, T), \qquad (7.1b)$$

$$\frac{\partial \mathbf{P}}{\partial t} = \mathbf{S}, \quad \frac{\partial \mathbf{Q}}{\partial t} = \mathbf{R} \qquad \text{in } \Omega \times (0, T), \qquad (7.1c)$$

$$E = -E^I \qquad \text{in } \partial \Omega_S \times (0, T), \qquad (7.1d)$$

$$E = 0, \quad \mathbf{H}_\perp = \mathbf{0} \qquad \text{in } \Omega \times \{0\}, \qquad (7.1e)$$

$$\mathbf{P} = \mathbf{0}, \quad \mathbf{Q} = \mathbf{0} \qquad \text{in } \Omega \times \{0\}, \qquad (7.1f)$$

where

$$\mathbf{P} = [P_x, P_y]^T, \qquad \mathbf{L} = \mathbf{A}\mathbf{H}_\perp + \mathbf{B}\mathbf{P}, \qquad \mathbf{S} = \mathbf{D}\mathbf{H}_\perp,$$

$$\mathbf{Q} = [Q_x, Q_y]^T, \qquad \mathbf{R} = \mathbf{G}\mathbf{Q} + \mathbf{F}\mathbf{H}_\perp, \qquad M = \mathbf{C}^T \mathbf{Q},$$

$$\mathbf{A} = \begin{bmatrix} -2\sigma_x & 0 \\ 0 & -2\sigma_y \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \sigma_x & 0 \\ 0 & -\sigma_y \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} -\frac{d\sigma_x}{dx} \\ \frac{d\sigma_y}{dy} \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} -\sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} -\sigma_x & 0 \\ 0 & -\sigma_y \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

with σ_x and σ_y, the defining damping property of the PML layer, given by

$$\sigma_x = \begin{cases} 0 & |x| < 1 \\ 15(x-1)^2 & x \geq 1 \\ 15(x+1)^2 & x \leq -1 \end{cases}, \quad \sigma_y = \begin{cases} 0 & |y| < 1 \\ 15(y-1)^2 & y \geq 1 \\ 15(y+1)^2 & y \leq -1 \end{cases}.$$

A typical truncated domain $[-1, 1]^2$ (fine mesh) together with the PML domain (coarse mesh) is shown in Figure 7.4. The object in the middle of the domain is a scatterer. We also show a typical scattered electric field solution in Figure 7.5.
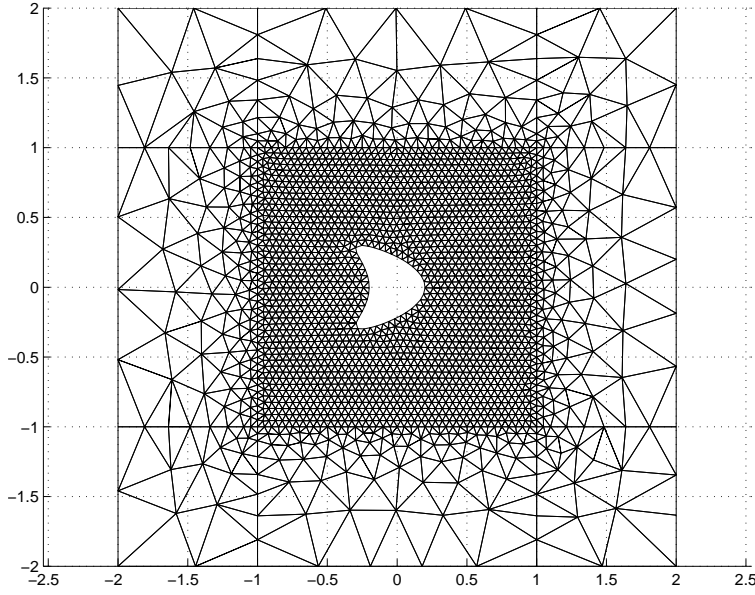
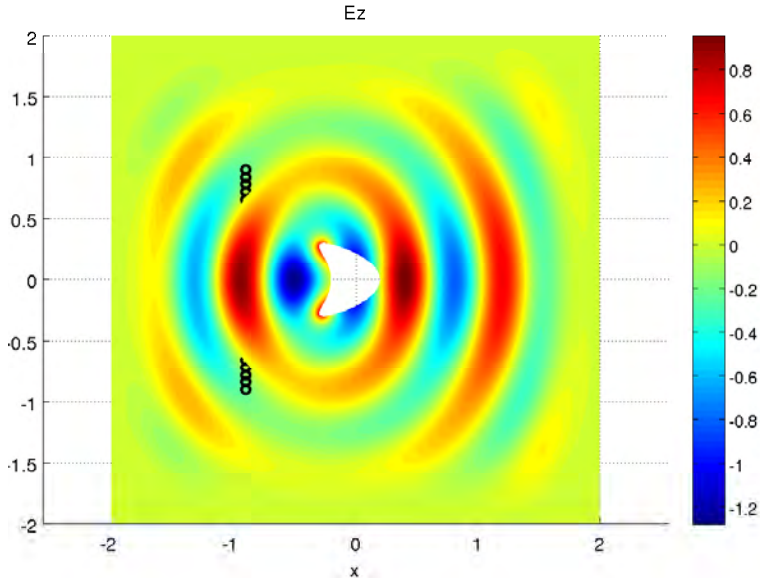FIG. 7.4. *A typical mesh of the PML and truncated domains together with a scatterer.*

FIG. 7.5. *An example of scattered electric field.*

The forward problem can be stated as follows. Given a scatterer's shape, the goal is to compute the scattered fields, in particular, at the observation points denoted as small circles in Figure 7.5.

In the inverse problem, on the other hand, the task is to reconstruct the scatterer's

shape, Ω_S, given scattered field data, possibly polluted by noises, at K observation points. It is not necessary to have the scattered data for all the fields, but we assume it is so for convenience. We choose to solve the inverse problem statistically using a Bayesian framework whose details can be found in [15, 35]. Assuming i.i.d. Gaussian noise with zero mean and variance σ^2 at all observation points, the likelihood model is chosen as

$\pi_{\text{like}} \propto$

$$\exp\left\{ -\frac{1}{2\sigma^2 T} \sum_{k=1}^{K} \int_T \int_\Omega \left[\left(E - E_k^{obs} \right)^2 + \left(\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs} \right)^T \left(\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs} \right) \right] \delta_{(\mathbf{x}-\mathbf{x}_k)} d\Omega dt \right\},$$

where quantities with superscript "obs" are the observed data and $\mathbf{x} = [x, y]^T$. The Dirac delta function $\delta_{(\cdot)}$ is defined as

$$\delta_{(\mathbf{x}-\mathbf{x}_k)} = \left\{ \begin{array}{ll} 1 & \mathbf{x} = \mathbf{x}_k, \\ 0 & \text{otherwise.} \end{array} \right.$$

We begin the prior modeling by defining the admissible shape space. In this paper, the shape parametrization is restricted as

$$r = \sum_{i=1}^{n_s} m^i \cos\left([i-1]\theta \right)$$

where (r, θ) are polar coordinates of the shape, and m^i the ith shape parameter. Assume *a priori* that the unknown shape is smooth so that the following spline smoothing can be employed [57]:

$$\pi_{\text{prior}} \propto \exp\left\{ -\frac{\kappa}{2} \int_0^{2\pi} \left(\frac{d^2 r}{d\theta^2} \right)^2 d\theta \right\}.$$

The solution to the statistical inverse problem is the following posterior density, after ignoring the normalized constant (which is not required by Markov chain Monte Carlo methods),

$$d\left(\mathbf{m}\right) = \pi_{\text{like}} \times \pi_{\text{prior}}.$$

Denote $\mathcal{J} = -\log d\left(\mathbf{m}\right)$, then we have

$$\mathcal{J} = \frac{1}{2\sigma^2 T} \sum_{k=1}^{K} \int_T \int_\Omega \left[\left(E - E_k^{obs} \right)^2 + \left(\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs} \right)^T \left(\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs} \right) \right] \delta_{(\mathbf{x}-\mathbf{x}_k)} d\Omega dt$$

$$+ \frac{\kappa}{2} \int_0^{2\pi} \left(\frac{d^2 r}{d\theta^2} \right)^2 d\theta.$$

One of the key ingredients of our approach is the Hessian (or its approximation) of $\mathcal{J}$. For large-scale PDE-based inverse problem such as the inverse scattering example considered in this section, efficient method for computing the Hessian is vital and we

adopt an adjoint approach to fulfill this goal. To begin, we form the Lagrangian

$$
\mathcal{L} = \mathcal{J} + \int_T \int_\Omega \mathbf{h}_\perp \cdot \left(\frac{\partial \mathbf{H}_\perp}{\partial t} + \nabla E - \mathbf{A} \mathbf{H}_\perp - \mathbf{B} \mathbf{P} \right) d\Omega dt
$$
$$
+ \int_T \int_\Omega e \left(\frac{\partial E}{\partial t} + \nabla \cdot \mathbf{H}_\perp - \mathbf{C}^T \mathbf{Q} \right) d\Omega dt + \int_T \int_\Omega \mathbf{p} \cdot \left(\frac{\partial \mathbf{P}}{\partial t} - \mathbf{D} \mathbf{H}_\perp \right) d\Omega dt
$$
$$
+ \int_T \int_\Omega \mathbf{q} \cdot \left(\frac{\partial \mathbf{Q}}{\partial t} - \mathbf{G} \mathbf{Q} - \mathbf{F} \mathbf{H}_\perp \right) d\Omega dt + \int_T \int_{\partial \Omega_S} \lambda \left(E + E^I \right) ds dt
$$
$$
+ \int_\Omega \mathbf{h}_I \cdot \mathbf{H}_\perp d\Omega + \int_\Omega \mathbf{p}_I \cdot \mathbf{P} d\Omega + \int_\Omega \mathbf{q}_I \cdot \mathbf{Q} d\Omega + \int_\Omega e_I E d\Omega.
$$

The first order Karush-Kuhn-Tucker optimality system can be derived as follows:
- Taking the first variation of the Lagrangian with respect to $\mathbf{h}_\perp, e, \mathbf{p}, \mathbf{q}$ and arguing that the variations of $\mathbf{h}_\perp, e, \mathbf{p}, \mathbf{q}$ are arbitrary in $\Omega \times (0, T)$ yield the forward equations (7.1a)– (7.1c).
- Taking the first variation of the Lagrangian with respect to λ and arguing that the variation of λ is arbitrary in $\partial \Omega_S \times (0, T)$ yield the forward PEC condition (7.1d).
- Taking the first variation of the Lagrangian with respect to $e_I, \mathbf{h}_I, \mathbf{p}_I, \mathbf{q}_I$ and arguing that the variations of $e_I, \mathbf{h}_I, \mathbf{p}_I, \mathbf{q}_I$ are arbitrary in $\Omega \times \{0\}$ yield the forward initial conditions (7.1e)–(7.1f).
- Taking the first variation of the Lagrangian with respect to $\mathbf{H}_\perp, E, \mathbf{P}, \mathbf{Q}$ and arguing that the variations of $\mathbf{H}_\perp, E, \mathbf{P}, \mathbf{Q}$ are arbitrary in the corresponding domains yield the following adjoint equations together with the final and boundary conditions

$$
\frac{\partial \mathbf{h}_\perp}{\partial t} + \nabla e = \mathbf{L}^* \qquad\qquad \text{in } \Omega \times (0, T), \qquad (7.2a)
$$

$$
\frac{\partial e}{\partial t} + \nabla \cdot \mathbf{h}_\perp = M^* \qquad\qquad \text{in } \Omega \times (0, T), \qquad (7.2b)
$$

$$
\frac{\partial \mathbf{p}}{\partial t} = -\mathbf{B}^T \mathbf{h}_\perp, \quad \frac{\partial \mathbf{q}}{\partial t} = \mathbf{G}^T \mathbf{q} - e\mathbf{C} \qquad \text{in } \Omega \times (0, T), \qquad (7.2c)
$$

$$
e = 0 \qquad\qquad \text{in } \partial \Omega_S \times (0, T), \qquad (7.2d)
$$

$$
e = 0, \quad \mathbf{h}_\perp = \mathbf{0} \qquad\qquad \text{in } \Omega \times \{T\}, \qquad (7.2e)
$$

$$
\mathbf{p} = \mathbf{0}, \quad \mathbf{q} = \mathbf{0} \qquad\qquad \text{in } \Omega \times \{T\}, \qquad (7.2f)
$$

where

$$
\mathbf{L}^* = \sum_{k=1}^K \left(\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs} \right) \delta_{(\mathbf{x} - \mathbf{x}_k)} - \mathbf{A}^T \mathbf{h}_\perp - \mathbf{D}^T \mathbf{p} - \mathbf{F}^T \mathbf{q},
$$
$$
M^* = \sum_{k=1}^K \left(E - E_k^{obs} \right) \delta_{(\mathbf{x} - \mathbf{x}_k)}.
$$

Other adjoint variables are found to be

$$
\lambda = -\mathbf{h}_\perp \cdot \mathbf{n} \qquad \text{in } \partial \Omega_S \times (0, T),
$$
$$
e_I = e, \quad \mathbf{h}_I = \mathbf{h}_\perp, \quad \mathbf{p}_I = \mathbf{p}, \quad \mathbf{q}_I = \mathbf{q} \qquad \text{in } \Omega \times \{0\}.
$$

- Taking derivatives of the Lagrangian with respect to the shape parameters m^i can be done using the shape gradient and Hessian methods as in [19]. Here, we assume that the obstacle is star-like around the origin and hence simpler route is possible (see [12] and references therein). The derivatives of the Lagrangian with respect to m^i turns out to be

$$\mathcal{G}_i = \frac{\partial \mathcal{L}}{\partial m^i} = -\int_T \int_0^{2\pi} \left[\mathbf{h}_\perp \cdot \nabla \left(E + E^I \right) \right] r \cos\left([i-1]\,\theta \right) d\theta dt. \qquad (7.3)$$

The reduced gradient computation at a particular shape $\mathbf{m}$ is now ready. One first solves the forward system (7.1a)–(7.1f) for the forward states and forward PML variables. The adjoint system (7.2a)–(7.2f) is then solved for the adjoint states and adjoint PML variables. The reduced shape gradient is now available by evaluating the right hand side of equation (7.3). It is clear that one forward and one adjoint solves are needed for the shape gradient computation.

In order to compute the product of the shape Hessian and a vector of shape variation, we first compute the forward variation, involving one incremental forward (linearization of the forward equations) solve, and then the adjoint variation, involving one incremental adjoint (linearization of the adjoint equations) solve, corresponding to that shape variation vector. The shape Hessian-vector product is the total variation of the shape gradient (7.3). More specifically, the variation δr due to variation $\delta \mathbf{m}$ is given by

$$\delta r = \sum_{i=1}^{n_s} \delta m^i \cos\left([i-1]\theta \right).$$

The ith component of the product of the shape Hessian and vector $\delta \mathbf{m}$ can be shown to be

$$\begin{aligned}
\delta G_i = &- \int_T \int_0^{2\pi} \nabla \left[\mathbf{h}_\perp \cdot \nabla \left(E + E^I \right) \right] \cdot \mathbf{e} r \cos\left([i-1]\,\theta \right) \delta r \, d\theta dt \\
&- \int_T \int_0^{2\pi} \left[\mathbf{h}_\perp \cdot \nabla \left(E + E^I \right) \right] \delta r \cos\left([i-1]\,\theta \right) d\theta dt \\
&- \int_T \int_0^{2\pi} \left[\delta\mathbf{h}_\perp \cdot \nabla \left(E + E^I \right) \right] r \cos\left([i-1]\,\theta \right) d\theta dt \\
&- \int_T \int_0^{2\pi} \left[\mathbf{h}_\perp \cdot \nabla \delta E \right] r \cos\left([i-1]\,\theta \right) d\theta dt,
\end{aligned}$$

where $\mathbf{e} = [\cos\theta, \sin\theta]^T$, and the variations in the forward states, δE and $\delta \mathbf{H}_\perp$, satisfy the following incremental forward equations

$$\begin{aligned}
\frac{\partial \delta \mathbf{H}_\perp}{\partial t} + \nabla \delta E &= \delta \mathbf{L} && \text{in } \Omega \times (0, T), \\
\frac{\partial \delta E}{\partial t} + \nabla \cdot \delta \mathbf{H}_\perp &= \delta M && \text{in } \Omega \times (0, T), \\
\frac{\partial \delta \mathbf{P}}{\partial t} = \delta \mathbf{S}, \quad \frac{\partial \delta \mathbf{Q}}{\partial t} &= \delta \mathbf{R} && \text{in } \Omega \times (0, T), \\
\delta E &= -\nabla \left(E + E^I \right) \cdot \mathbf{e} \, \delta r && \text{in } \partial\Omega_S \times (0, T), \\
\delta E = 0, \quad \delta \mathbf{H}_\perp &= \mathbf{0} && \text{in } \Omega \times \{0\}, \\
\delta \mathbf{P} = \mathbf{0}, \quad \delta \mathbf{Q} &= \mathbf{0} && \text{in } \Omega \times \{0\},
\end{aligned}$$

where

$$\delta\mathbf{P} = [\delta P_x, \delta P_y]^T, \qquad \delta\mathbf{L} = \mathbf{A}\delta\mathbf{H}_\perp + \mathbf{B}\delta\mathbf{P}, \qquad \delta\mathbf{S} = \mathbf{D}\delta\mathbf{H}_\perp,$$
$$\delta\mathbf{Q} = [\delta Q_x, \delta Q_y]^T, \qquad \delta\mathbf{R} = \mathbf{G}\delta\mathbf{Q} + \mathbf{F}\delta\mathbf{H}_\perp, \qquad \delta M = \mathbf{C}^T\delta\mathbf{Q}.$$

Similarly, the variations in the adjoint states, δe and $\delta\mathbf{h}_\perp$, satisfy the following incremental adjoint equations

$$\frac{\partial\delta\mathbf{h}_\perp}{\partial t} + \nabla\delta e = \delta\mathbf{L}^* \qquad\qquad \text{in } \Omega \times (0,T),$$
$$\frac{\partial\delta e}{\partial t} + \nabla\cdot\delta\mathbf{h}_\perp = \delta M^* \qquad\qquad \text{in } \Omega \times (0,T),$$
$$\frac{\partial\delta\mathbf{p}}{\partial t} = -\mathbf{B}^T\delta\mathbf{h}_\perp, \quad \frac{\partial\delta\mathbf{q}}{\partial t} = \mathbf{G}^T\delta\mathbf{q} - \delta e\mathbf{C} \qquad \text{in } \Omega \times (0,T),$$
$$\delta e = -\nabla e\cdot\mathbf{e}\,\delta r \qquad\qquad \text{in } \partial\Omega_S \times (0,T),$$
$$e = 0, \quad \delta\mathbf{h}_\perp = \mathbf{0} \qquad\qquad \text{in } \Omega \times \{T\},$$
$$\delta\mathbf{p} = \mathbf{0}, \quad \delta\mathbf{q} = \mathbf{0} \qquad\qquad \text{in } \Omega \times \{T\},$$

where

$$\delta\mathbf{L}^* = \sum_{k=1}^K \delta\mathbf{H}_\perp\delta_{(\mathbf{x}-\mathbf{x}_k)} - \mathbf{A}^T\delta\mathbf{h}_\perp - \mathbf{D}^T\delta\mathbf{p} - \mathbf{F}^T\delta\mathbf{q},$$

$$\delta M^* = \sum_{k=1}^K \delta E\delta_{(\mathbf{x}-\mathbf{x}_k)}.$$

During the Newton iterations in which the shape is updated, we generate a new corresponding mesh for the (incremental) forward and (incremental) adjoint solves. To simplify the implementation and to avoid difficulties for the mesh generator, we allow the shape parameters to vary only in the hyper-rectangle defined by

$$\mathbf{m}_L = 0.15\left[1, -1, -1/2, \ldots, -1/2^{(n_s-1)}\right]^T,$$
$$\mathbf{m}_U = \left[0.27, 0.15, 0.15/2, \ldots, 0.15/2^{(n_s-1)}\right]^T. \tag{7.4}$$

As a consequence, we fix the time step and hence avoid interpolating the solutions in time during the optimization process.

We use a 2nd-order nodal discontinuous Galerkin method [29] for spatial discretization and the classical 4th-order Runge-Kutta for temporal discretization of the (incremental) forward and (incremental) adjoint equations. The mesh has 4,494 triangles and the total number of nodal unknowns, for both $\mathbf{H}_\perp$ and E, is 80,892. The observation data E^{obs} and $\mathbf{H}_\perp^{obs}$ are synthesized by solving the forward solver $T = \pi$ using 4474 time steps. The exact shape that we would like to invert for is governed by

$$x = \cos(t) + 0.65\cos(2t) - 0.65, \quad y = 1.5\sin(t), \quad t \in [0, 2\pi].$$

This exact shape and the corresponding electric field at $T = \pi$ are shown in Figure 7.5. For convenience, we compute the observations at all time steps and add an i.i.d. zero mean Gaussian noise with $\sigma = 0.05$. The regularization parameter κ is chosen to be $1/(\sigma^2 T)$.

For the first greedy cycle, we use the circle with radius of 0.236 as the initial guess for the optimization solver. For other greedy cycles, the initial guesses are computed, as in Section 6, from the test set $\boldsymbol{M}^a$ containing 10000 LHC points distributed within the bound $\mathbf{m}_L \leq \mathbf{m} \leq \mathbf{m}_U$.

Next, we use DRAM, an efficient MCMC package in [28], to sample our Gaussian process predictor with one million MCMC simulations. Figure 7.6 shows the result for 29 training points and one global Gaussian approximation after one greedy cycle. Figure 7.7 shows the result for 62 training points and six local Gaussian approximations after ten greedy cycles. Here, we plot the sample posterior mean and its 99.99% credibility envelope together with the exact shape and the deterministic solution. As can be seen, the posterior mean predicts well the left side of the kite but worse on the right. This is expected since the incident wave is from left to right. The credibility region responses similarly, namely, the uncertainty is less on the left and grows gradually as we move to the right. Nevertheless, the uncertainty is large, even on the left side, at the center of the concave part of the kite. This is anticipated since non-convex regions are not easy to be reconstructed [38]. It is interesting to observe that within the bounds of interest (7.4), the approximations with one and ten greedy cycles are not very different. It suggests that there be a single dominant mode of the Bayesian posterior density which is already captured by the first greedy cycle. One can also see that the sample mean is almost identical to the deterministic solution.
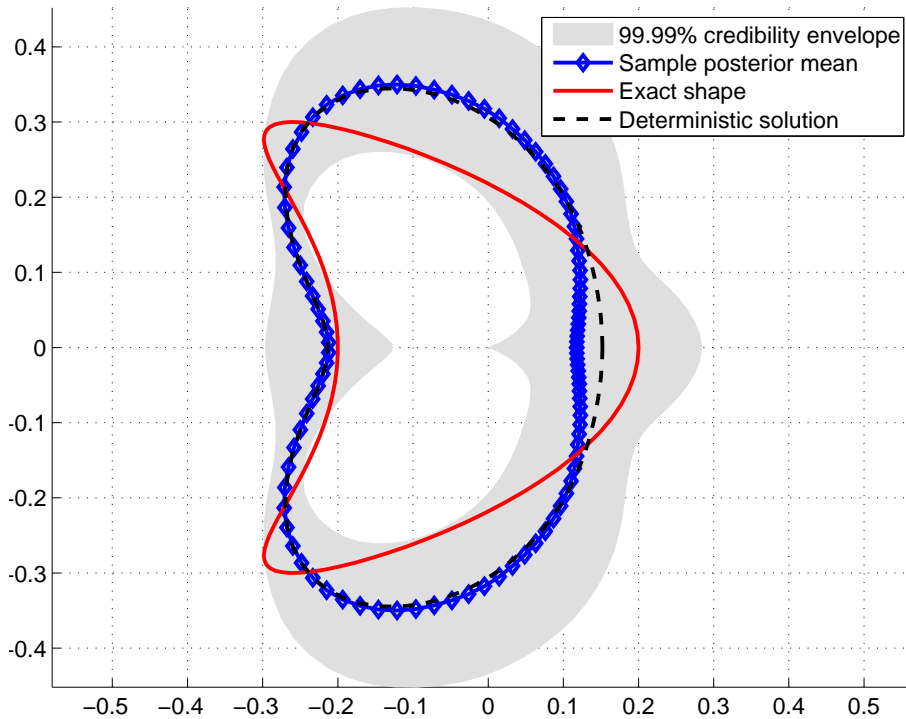

FIG. 7.6. *The sample posterior mean from one million MCMC simulations together with its 99.99% credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after one greedy cycle accounting for 29 training points and one global Gaussian approximation.*
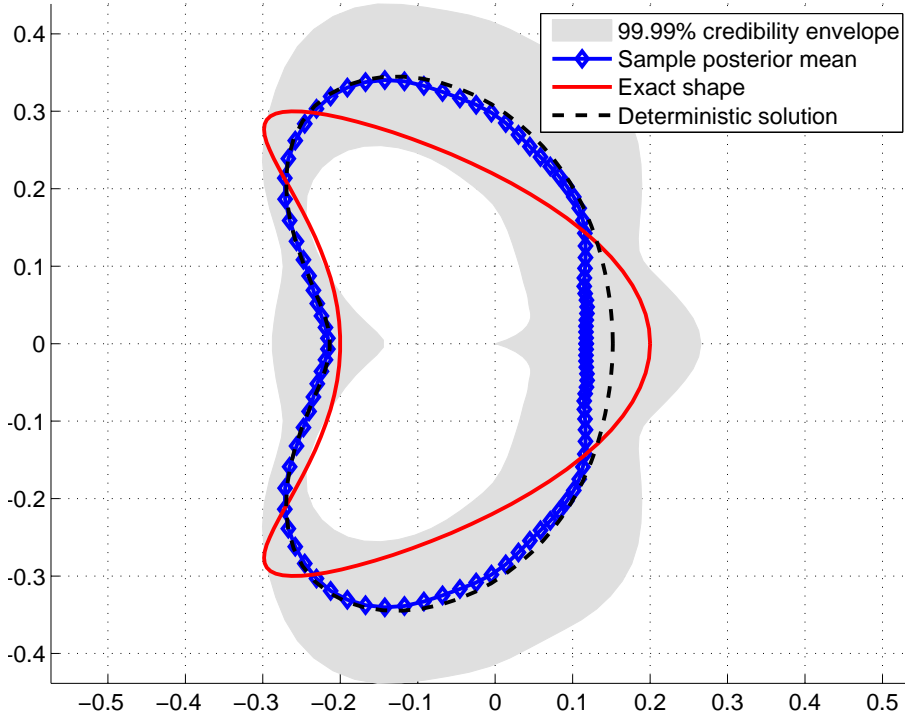
FIG. 7.7. *The sample posterior mean from one million MCMC simulations together with its 99.99% credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after ten greedy cycle accounting for 62 training points and one global Gaussian approximation.*

It should be pointed out that a million MCMC simulations is chosen randomly, but it turns out that the MCMC already converges. To see this, we perform two million MCMC simulations and show again the sample posterior mean together with its 99.99% credibility envelope versus the exact and the deterministic solutions in Figures 7.8 and 7.9. The results look almost unchanged compared to those in Figures 7.6 and 7.7. To further confirm the convergence, we plot two dimensional marginal chains for the Gaussian process predictor obtained after ten greedy cycle using one million MCMC simulations in Figures 7.10 and 7.11, and two million MCMC simulations in Figures 7.12 and 7.13. As can be observed, the marginal chains (only the first eight parameters are shown) converge and look almost indifferent for both MCMC runs.

To see the efficiency of the Gaussian response surface method, we compare the CPU time taken for two million MCMC simulations for one greedy cycle case in Table 7.6. The offline time is defined as the time taken to build the Gaussian process response surface. It turns out that it took about 33 hours, while there would have been no offline cost if we had used the exact Bayesian posterior density. However, the offline time was paid off when we performed the MCMC simulations. In particular, the MCMC simulations required almost one hour for the GP response surface, but it would have taken 528141 hours if we had used the exact posterior density!

8. Conclusions. We have developed an adaptive Hessian-based non-stationary Gaussian process response surface method to approximate a probability density func-

TABLE 7.6
CPU time taken for two million MCMC simulations.

	Gaussian process predictor	exact posterior density
offline time	33 hours	0 hours
online time	0.96 hours	528141.75 hours

tion (pdf) that exploits its structure, in particular the Hessian of its negative loga-
rithm. Of particular interest to us are expensive-to-evaluate pdfs, e.g., those arising
from the Bayesian solution of large-scale inverse problems. Our method can be con-
sidered as a piecewise adaptive Gaussian approximation in which a Gaussian tailored
to the local Hessian of the negative log probability density is constructed for each
sub-region in high dimensional parameter space. The task of efficiently partition-
ing the parameter space into sub-regions is done implicitly through Hessian-informed
membership probability functions. The Gaussian process machinery is then employed
to glue all local Gaussian approximations into a global analytical response surface
that is far cheaper to evaluate than the original expensive probability density. The
resulting response surface is also equipped with an analytical variance estimate that
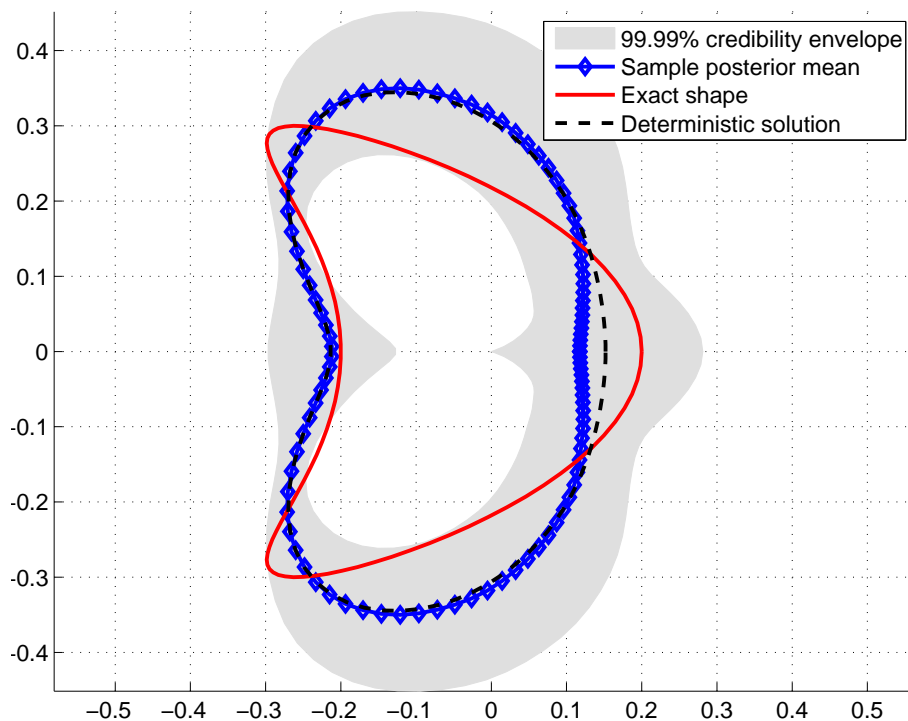can be used to assess the uncertainty of the approximation. One of the key com-

FIG. 7.8. *The sample posterior mean from two million MCMC simulations together with its
99.99% credibility envelope versus the exact and the deterministic solutions. The Gaussian predic-
tor is obtained after one greedy cycle accounting for 29 training points and one global Gaussian
approximation.*

ponents of our proposed approach is an adaptive sampling strategy for exploring the
parameter space efficiently during the computer experimental design step, which aims
to find training points with high probability density. The detailed construction and
an analysis of the method have been presented. We have demonstrated the accuracy
and efficiency of the proposed method on several example problems, including inverse
shape electromagnetic scattering in 24-dimensional parameter space.

Ongoing research aims to address the following:

1. The Gaussian process predictor is not guaranteed to be non-negative every-
   where (even though it seems to be the case for all examples considered in this
   paper). How to enforce the positiveness of the predictor in our framework
   remains an open question, though one may use an approach proposed in [52].
2. Rigorous analysis of the quality of the Gaussian process predictor is clearly
   an important direction for future work.
3. The size of the random set $\mathbf{M}^a$ is quite arbitrary. Intuitively, the larger it is,
   the better our predictor. Thus, the question that needs to be addressed is
   how to choose $\mathbf{M}^a$ as a function of the dimension of the parameter space.
4. Constructing the full Hessian for local Gaussian approximation is prohibitively
   expensive for practical inverse problems in high dimensional parameter spaces.
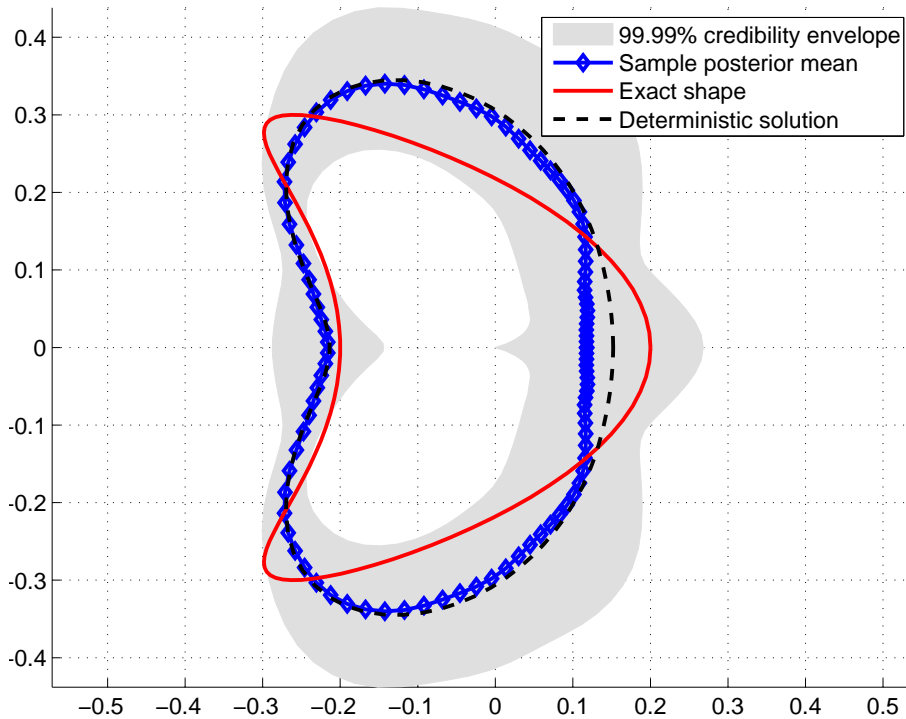   However, often, one can make accurate low rank approximations of the Hes-

FIG. 7.9. *The sample posterior mean from two million MCMC simulations together with its
99.99% credibility envelope versus the exact and the deterministic solutions. The Gaussian pre-
dictor is obtained after ten greedy cycle accounting for 62 training points and one global Gaussian
approximation.*

sian for ill-posed inverse problem; scalable algorithms can be constructed for this task [22] and theoretical justification of the compactness of the Hessian can be provided in certain cases [12, 13].

5. We have concentrated on the detailed development and analysis of the proposed method, and on its verification for several examples. We have compared our approach only to the popular adaptive radial basis function method. Our future work should carry out extensive comparisons with other existing methods as well.

6. The adaptive Hessian-based non-stationary Gaussian process response surface method is designed for problems involving thousands of parameters or more. Ongoing research is to apply the method to such high-dimensional problems as well.
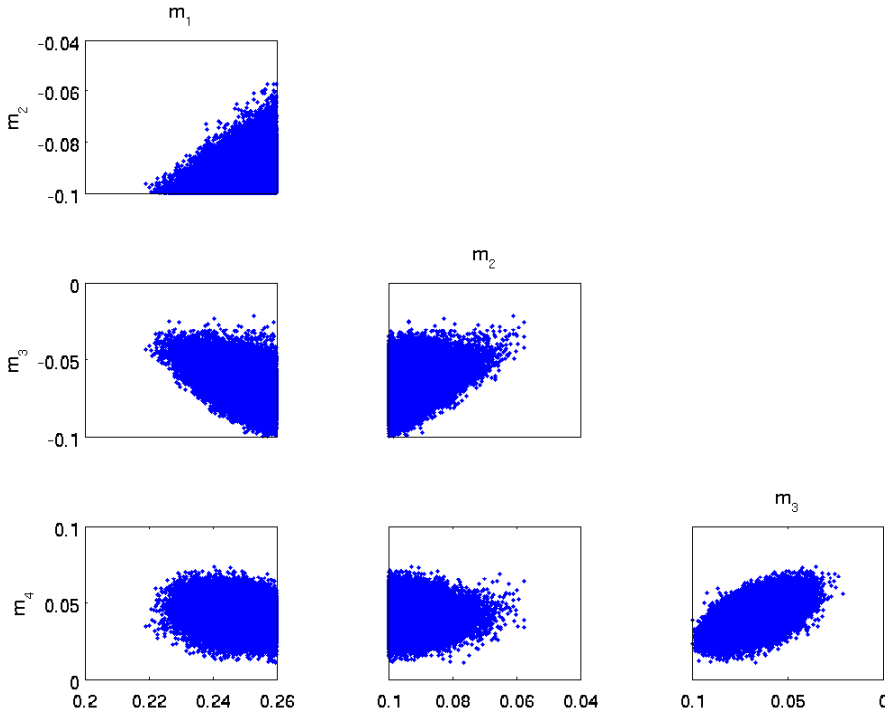
REFERENCES

FIG. 7.10. *Two dimensional marginal chains for parameters m^1, m^2, m^3, m^4. The Gaussian process predictor is obtained after ten greedy cycle using one million MCMC simulations.*

[1] SAUL ABARBANEL AND DAVID GOTTLIEB, *On the construction and analysis of absorbing layers in CEM*, Applied Numerical Mathematics, 27 (1998), pp. 331–340.

[2] PETTER ABRAHAMSEN, *A review of Gaussian random fields and correlation functions*, Tech. Report 2nd edition, Norwegian Computing Center, Box 114, Blindern N-0314 Oslo, Norway, 1997.

[3] VOLKAN AKÇELIK, JACOBO BIELAK, GEORGE BIROS, IOANNIS EPANOMERITAKIS, ANTONIO FERNANDEZ, OMAR GHATTAS, EUI J. KIM, JULIO LOPEZ, DAVID R. O'HALLARON, TIANKAI TU, AND JOHN URBANIC, *High resolution forward and inverse earthquake modeling on terascale computers*, in SC03: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, ACM/IEEE, 2003. Gordon Bell Prize for Special Achievement.

[4] VOLKAN AKÇELIK, GEORGE BIROS, OMAR GHATTAS, JUDITH HILL, DAVID KEYES, AND BART VAN BLOEMAN WAANDERS, *Parallel PDE-constrained optimization*, in Parallel Processing for Scientific Computing, M. Heroux, P. Raghaven, and H. Simon, eds., SIAM, 2006.

[5] RICHARD BELLMAN, *Adaptive control processes: A guided tou*, Princeton Univ. Press, 1961.

[6] G. BIROS AND O. GHATTAS, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. Part I: The Krylov–Schur solver*, SIAM Journal on Scientific Computing, 27 (2005), pp. 687–713.

[7] ———, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. Part II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows*, SIAM Journal on Scientific Computing, 27 (2005), pp. 714–739.

[8] S. BOYAVAL, C. LE BRIS, T. LELIEVRE, Y. MADAY, N.C. NGUYEN, AND A. T. PATERA, *Reduced basis techniques for stochastic problems*, (2009). Submitted to Archives of Computational Methods in Engineering.

[9] J. S. BRIDLE, *Probabilistic interpretation of feedforward classification network ouptuts, with relationship to statistical pattern recognition*, Neuro-computing: algorithms, architectures and applications, Spinger-Verlag, 1989.
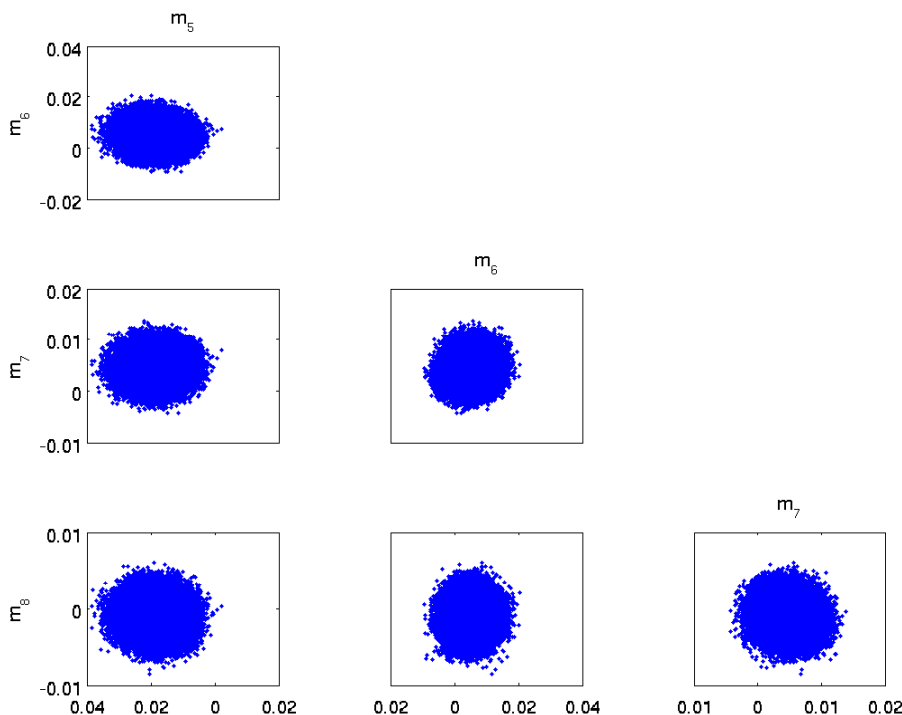

FIG. 7.11. *Two dimensional marginal chains for parameters m^5, m^6, m^7, m^8. The Gaussian process predictor is obtained after ten greedy cycle using one million MCMC simulations.*
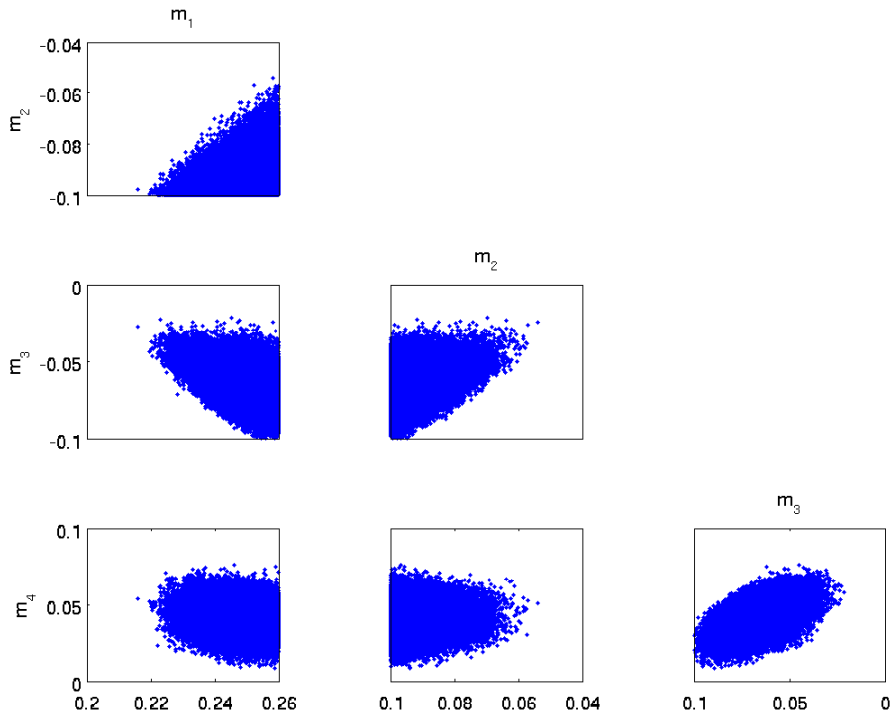
FIG. 7.12. *Two dimensional marginal chains for parameters m^1, m^2, m^3, m^4. The Gaussian process predictor is obtained after ten greedy cycle using two million MCMC simulations.*

[10] M. D. Buhmann, *Radial basis functions: theory and implementations*, vol. 12 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.

[11] Tan Bui-Thanh, *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*, PhD thesis, MIT, 2007.

[12] Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part I: Inverse shape scattering of acoustic waves*, submitted, (2011).

[13] ———, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, submitted, (2011).

[14] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3270–3288.

[15] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, Springer, New York, 2007.

[16] Sung-Hyuk Cha, *Comprehensive survey on distance/similarity measures between probability density functions*, International Journal of Mathematical Models and Methods in Applied Sciences, 1 (2007), pp. 300–307.

[17] Andres Christen and Bruno Sanso, *Advances in the design of Gaussian processes as surrogate models for computer experiments*, Tech. Report 5, University of California, Santa Cruz, 2008.

[18] David A. Cohn, *Neural networks exploration using optimal experiment design*, Neural Networks, 9 (1996), pp. 1071–1083.

[19] M. C. Delfour and J.-P. Zolésio, *Shapes and Geometries. Analysis, Differential Calculus, and Optimization*, Advances in Design and Control, SIAM, Philadelphia, 2001.

[20] David L. Donoho, *High-dimensional data analysis: The curses and blessings of dimensionality.* Lecture at American Math. Society "Math Challenges of the 21st Century", 2000.

[21] I. Epanomeritakis, V. Akçelik, O. Ghattas, and J. Bielak, *A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion*, Inverse Problems,
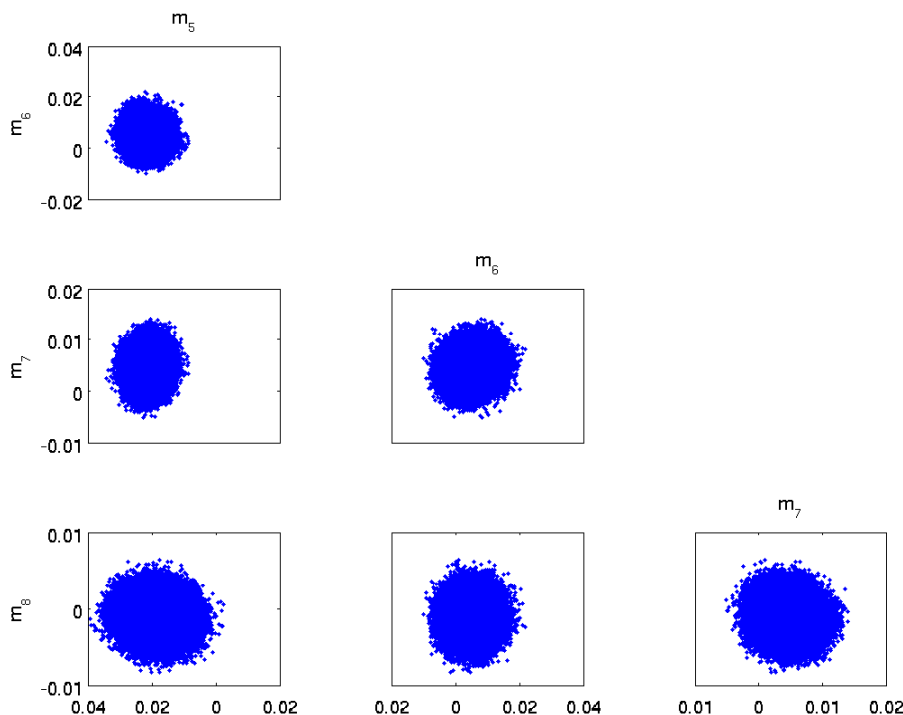
FIG. 7.13. *Two dimensional marginal chains for parameters m^5, m^6, m^7, m^8. The Gaussian process predictor is obtained after ten greedy cycle using two million MCMC simulations.*

24 (2008), p. 034015 (26pp).

[22] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, *Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations*, SIAM Journal on Scientific Computing, 33 (2011), pp. 407–432.

[23] Montserrat Fuentes and Richard L. Smith, *A new class of nonstationary spatial models*, tech. report, North Carolina state University, 2001.

[24] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas, *Nonlinear model reduction for uncertainty quantification in large-scale inverse problems*, International Journal For Numerical Methods in Engineering, 81 (2010), pp. 1581–1608.

[25] Mark N. Gibbs, *Bayesian Gaussian Processes for regression and classification*, PhD thesis, Cambridge University, 1997.

[26] Robert R. Gramacy, Her K.H. Lee, and William MacReady, *Parameter space exploration with gaussian process trees*, in 21st international conference on Machine Learning, 2004.

[27] Martin A. Grepl and Anthony T. Patera, *A posteriori error bounds for reduced-bias approximations of parametrized parabolic partial differential equations*, M2AN. Mathematical Modelling and Numerical Analysis, 39 (2005), pp. 157–181.

[28] Heikki Haario, Marko Laine, Antonietta Miravete, and Eero Saksman, *DRAM: Efficient adaptive MCMC*, Statistics and Computing, 16 (2006), pp. 339–354.

[29] Jan S. Hesthaven and Timothy Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, vol. 54 of Texts in Applied Mathematics, Springer, 2008.

[30] David Higdon, James Gattiker, Brian Williams, and Maria Rightley, *Computer model calibration using high-dimensional output*, Journal of the American Statistical Association, 103 (2008), pp. 570–583.

[31] David Higdon, Marc Kennedy, James C. Cavendish, John A. Cafeo, and Robert D. Ryne, *Combining field data and computer simulations for calibration and prediction*, SIAM Journal on Scientific Computing, 26 (2004), pp. 448–466.

[32] DAVID HIGDON, J. SWALL, AND J. KERN, *Nonstationary spatial modeling*, Bayesian statistics 6, Oxford University Press, 1999, pp. 761–768.

[33] M. HINZE, R. PINNAU, M. ULBRICH, AND S. ULBRICH, *Optimization with PDE Constraints*, Springer, 2009.

[34] D.B.P. HUYNH, D. J. KNEZEVIC, AND A. T. PATERA, *Certified reduced basis model characterization: a frequentistic uncertainty framework*, (2011). Submitted to Computer Methods in Applied Mechanics and Engineering.

[35] JARI KAIPIO AND ERKKI SOMERSALO, *Statistical and Computational Inverse Problems*, vol. 160 of Applied Mathematical Sciences, Springer-Verlag, New York, 2005.

[36] MARC C. KENNEDY AND ANTHONY O'HAGAN, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society. Series B (Statistical Methodology), 63 (2001), pp. 425–464.

[37] HYOUNG-MOON KIM, BANI K. MALLICK, AND C. C. HOLMES, *Analyzing nonstationary spatial data using piecewise Gaussian processes*, Journal of the American Statistical Association, 100(470) (2005), pp. 653–668.

[38] A. KIRSCH, R. KRESS, P. MONK, AND A. ZINN, *Two methods for solving the inverse acoustic scattering problem*, Inverse Problems, 4 (1988), pp. 749–770.

[39] ANDREAS KRAUSE, AJIT SINGH, AND CARLOS GUESTRIN, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, Journal of Machine Learning Research, 9 (2008), pp. 235–284.

[40] CHAD LIEBERMAN, KAREN WILLCOX, AND OMAR GHATTAS, *Parameter and state model reduction for large-scale statistical inverse problems*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2523–2542.

[41] DAVID J. C. MACKAY, *Information-based objective functions for active data selection*, Neural Computation, 4 (1992), pp. 590–604.

[42] ———, *Bayesian Methods for backpropagation networks*, Models of neural networks III, Springer-Verlag, 1994, ch. 6.

[43] ———, *Choice of basis for Laplace approximation*, Machine Learning, 33(1) (1998), pp. 77–86.

[44] JAMES MARTIN, LUCAS C. WILCOX, CARSTEN BURSTEDDE, AND OMAR GHATTAS, *A stochastic Newton MCMC method for large scale statistical inverse problems with application to seismic inversion*, submitted to SIAM Journal on Scientific Computing, (2011). submitted.

[45] Y. M. MARZOUK AND H. N. NAJM, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, Journal of Computational Physics, 228 (2009), pp. 1862–1902.

[46] YOUSSEF M. MARZOUK, HABIB N. NAJM, AND LARRY A. RAHN, *Stochastic spectral methods for efficient bayesian solution of inverse problems*, Journal of Computational Physics, 224 (2007), pp. 560 – 586.

[47] ANOOP A. MULLUR AND ACHILLE MESSAC, *Extended radial basis functions: more flexible and effective metamodeling*, American Institute of Aeronautics and Astronautics Journal, 43 (2005), pp. 1306–1315.

[48] RADFORD M. NEAL, *Bayesian learning for neural networks*, Spinger-Verlag, 1996.

[49] NGOC CUONG NGUYEN, *An uncertainty quantification method for parameter estimation in elliptic partial differential equations*, (2009). Submitted to Computer Methods in Applied Mechanics and Engineering.

[50] DAVID J. NOTT AND WILLIAM T. M. DUNSMUIR, *Estimation of nonstationary spatial covariance structure*, Biometrika, 89(4) (2002), pp. 819–829.

[51] DOUGLAS NYCHKA, CHRISTOPHER WIKLE, AND J. ANDREW ROYLE, *Large spatial prediction problems and nonstationary random field*, tech. report, Geophysical Statistical program, National center for atmospheric research, 1999.

[52] J. E. OAKLEY AND A. O'HAGAN, *Uncertainty in prior elicitations: a non–parametric approach*, Biometrika, 94 (2007), pp. 427–441.

[53] A. O'HAGAN, *Bayesian analysis of computer code outputs: A tutorial*, Reliability Engineering and System Safety, 91 (2006), pp. 1290–1300.

[54] CHRISTOPHER J. PACIOREK AND MARK J. SCHERVISH, *Spatial modelling using a new class of nonstationary covariance functions*, EnvironMetrics, 17 (2006), pp. 483–506.

[55] TOBIAS PFINGSTEN, MALTE KUSS, AND CARL E. RASMUSSEN, *Nonstationary Gaussian process regression using a latent extension of the input space*, in the ISBA Eighth World Meeting on Bayesian Statistics in Valencia, 2006.

[56] CARL E. RASMUSSEN AND CHRISTOPHER K. WILLIAMS, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[57] C. H. REINSCH, *Smoothing by spline functions*, Numerische Mathematik, 16 (1971), pp. 451–454.

[58] PETER D. SAMPSON AND PAUL. GUTTORP, *Nonparametric estimation of nonstationary spa-

tial covariance structure, Journal of the American statistical association, 87(417) (1992), pp. 108–119.

[59] Thomas J. Santner, Brian J. Williams, and William I. Notz, *The Design and Analysis of Computer Experiments*, Spinger-Verlag, 2003.

[60] Alexandra M. Schmidt and Anthony O'Hagan, *Bayesian inference for nonstationary spatial covariance structure via spatial derformation*, Journal of the Royal Statistical Society, Series B, 65 (2003), pp. 745–758.

[61] D. W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*, Wiley Series in Probability and Statistics, 1992.

[62] David W. Scott and Stephan R. Sain, *Multi-dimensional density estimation*, Handbook of Statistics, 24 (2005), pp. 229–261.

[63] Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer, *Gaussian process regression: active data selection and test point rejection*, Proceedings of the International Joint Conference on Neural Networks IJCNN 2000, IEEE, 2000, pp. 241–246.

[64] T. W. Simpson, J. D. Peplinski, P. N. Koch, and J K. Allen, *Metamodels for computerbased engineering design: Survey and recommendations*, Engineering with Computers, 17 (2001), pp. 129–150.

[65] Jingbo Wang and Nicholas Zabaras, *Using Bayesian statistics in the estimation of heat source in radiation*, International Journal of Heat and Mass Transfer, 48 (2005), pp. 15 – 29.